

AD-A148 247

CASH FLOW COMPUTER MODEL USERS GUIDE REVISION 1(U)  
OFFICE OF THE DEPUTY UNDER SECRETARY OF DEFENSE  
RESEARCH AND ENGINEERING (ACQUISITION MGMT) WASHINGTON

1/1

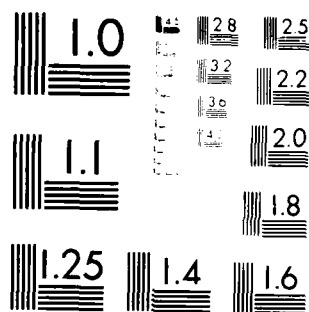
UNCLASSIFIED

DC 01 FEB 83

F/G 9/2

NL

END  
DATE  
FILMED  
DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A148 247



DEPARTMENT OF DEFENSE

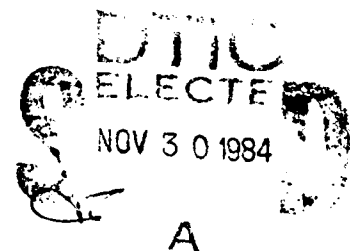
CASH FLOW COMPUTER MODEL

USERS GUIDE

REVISION 1

DTIC FILE COPY

*Supplemental*  
*AD A104091*  
*AD A129476*



CHAIRMAN, DOD CONTRACT FINANCE COMMITTEE  
OFFICE OF COST, PRICING & FINANCE  
OUSDRE (ACQUISITION MANAGEMENT)  
PENTAGON, WASHINGTON, D. C. 20301

FEBRUARY 1, 1983

This document has been approved  
for public release and sale; its  
distribution is unlimited.

84 11 28 057

## TABLE OF CONTENTS

1. INTRODUCTION	1
WHAT IS CASHII	1
WHY USE CASHII	2
SAMPLE CASHII RUN	4
HOW TO USE THE REST OF THE BOOK	10
2. CASHII BASICS	11
GETTING STARTED	11
CREATING A DATA FILE	13
SECTION 1	13
SECTION 2 (COST ELEMENTS)	14
SECTION 3 (TRANSACTIONS)	21
EXECUTING CASHII	29
3. CASHII OUTPUT	32
TRANSACTION REPORT	32
ANALYTICAL REPORT	35
RECAP OF COST STATEMENT REPORT	38
4. CASHII TERMS, DEFINITIONS & SYNTAX	39
TERMS AND DEFINITIONS	39
ERROR MESSAGES	49
5. APPENDICES	55
SAMPLE RUN USING GEISCO	
SAMPLE RUN USING BCS	
REFERENCE MATERIAL	



A1

## 1. INTRODUCTION

### WHAT IS CASHII

#### COMMERCIAL VENDORS

CASHII is a FORTRAN program which is available to be run on contractors own computers. Of course, even though the program is written as generically as possible, it is possible that some conversion effort would be required in order to make it execute correctly. Therefore, working versions of the program are maintained on two commercial computer service companies, General Electric Information Services and Boeing Computer Services. Contractors may elect to use the program on either of these companies. Contracting Officers use the program on the Copper Impact System (BCS). Contractors are not permitted to access the Copper Impact System. However, every effort is made to insure that the program is the same in all places.

#### VENDOR CONTACTS

Each vendor has identified a contact to assist in initial set up of the CASHII system. For BCS the name and address is:

M. Rosenberg  
BCS  
7980 Gallows Road  
Vienna, Va 22180  
(703) 821-~~62100~~  
6200

For GEISCo contact:

B. Clarke  
GEISCo  
1300 N. 17th Street - Suite 1100  
Arlington, Va. 22209  
(703) 276-4046

CASHII is the name of a computer program developed by the Department of Defense. Contractors who request flexible progress payments must use CASHII to provide the relevant calculations on flexible payment rate and contractor investment ratio. This data must also be supplied to contracting officers who use CASHII to verify the contractors data.

## 1. INTRODUCTION CONT'D

### WHY USE CASHII

-----

The following is excerpted from the Defense Acquisition Regulations Appendix E Part 5: E-530.1 Background and Policy. Paying progress payments assists in financing a contractor's performance and reduces the contractor's investment in its work in process inventory. The actual investment held by a contractor in work in process inventory is influenced by a number of factors in addition to progress payments, such as delivery schedules, cash management practices, and Government payment practices. Progress payment amounts that are determined by using uniform, standard progress payment rates (i.e., 90% or 95% under E-503) are insensitive to these other factors influencing investment and, as a consequence, result in investments by contractors in work in process inventory that vary among contractors and across contracts; on the other hand, flexible progress payment rates (expressed as a percentage that will be applied to costs to determine the amount payable as a progress payment rate) are designed to tailor more closely the progress payment rate to the cash needs for financing performance of a particular contractor for a given contract.

For flexible progress payments, cash needs are measured and projected in relation to investment underlying the work in process inventory over the life of the contract. Total investment is measured by a weighted average of total costs paid by the contractor to complete performance of the contract, and the contractor's investment is the weighted average of the amount not paid by the Government. The DOD, as a matter of policy, has concluded that a contractor should retain at least a 5% investment in work in process inventory over the life of the contract. Accordingly, the DOD will make progress payments at a rate (expressed as a whole number) that is the highest rate which yields a corresponding investment by the contractor in work in process inventory of not less than 5%. This progress payment rate is to be determined by the DOD Cash Flow Computer Model. In no event will the progress payment rate be greater than 100% or less than the uniform, standard progress payment rate that would have been applied to the contract absent flexible progress payments.

1. INTRODUCTION CONT'D

WHY USE CASHII

Contracting officers shall use a flexible progress payment rate in lieu of uniform, standard progress payment rates for the purpose of computing monthly progress payments if requested by the contractor, if the criteria in E-530.3 are met, and if the contractor agrees to the requirements of this section. Flexible progress payments shall be regarded as customary progress payments, as defined in E-503.

E-530.2 DOD Cash Flow Computer Model. The flexible progress payment rate shall be determined through application of the DOD Cash Flow Computer Model, available to contracting officers on the COPPER IMPACT computer time sharing network under the computer file name "CASHII". The model takes into account key cash flow factors, such as contract cost profile, delivery schedules, subcontractor progress payments, liquidation rates, and payment/reimbursement cycles. Operating instructions and cash flow data requirements are retrievable within the model in a conversational mode. Contractors may obtain copies of the DOD Cash Flow Computer Model User's Guide (including the related computer program) from the Defense Technical Information Center, Building 5 Cameron Station, Alexandria, Va 22314. Contracting officers may not grant contractor access to Government leased COPPER IMPACT time sharing computer network.

FLEXIBLE  
PAYMENT  
ELIGIBILITY

For negotiated fixed price type contracts over \$1 million, flexible progress payments may be authorized if requested by the contractor. The flexible progress payments may range from 90% to 100% as determined by the cash flow model which is the subject of this document.

## 1. INTRODUCTION CONT'D

### A SAMPLE CASHII RUN

-----

#### DATA FILE REQUIRED

Figures 1, 2 and 3 illustrate a simple run of the program. Figure 1 shows that the program always starts by asking you to answer six questions. These questions are asked at the terminal in order to provide you maximum flexibility. The rest of the data required by CASHII is in a file which the program reads. You are asked to supply the filename even before the six questions shown here. The first four questions are asked at the terminal so that the information contained in the data file can be kept constant while different answers are supplied here. This capability provides some "what if" analysis through CASHII.

#### RECAP OF COST STATEMENTS

The last two questions are asking how much detail you wish to see for output. A recap of the cost report data reformats and prints the data found in the input file. Printing this data can be very useful in verifying that the system is reading the data you think you typed.

#### TRANSACTION REPORT

A transaction report is a detailed report showing one line for every transaction that takes place. The system inserts transactions for planned progress payments. This enables you to analyze exactly how cash flow is effected by different assumptions.

#### ANALYTICAL REPORT

Both of these reports are discussed in detail in section 3 of this document. For now, however let's continue with the sample output shown in Figure 2. It is the Analytical Report. The Analytical Report is always produced, no matter how you answer the preceeding two questions.



\*\*\*\*\*  
\*\*\* DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL \*\*\*  
\*\*\* VERSION DATED FEBRUARY 1, 1983 \*\*\*  
\*\*\*\*\*

CAUTION--THIS CASH MODEL VERSION IS NOT COMPATIBLE WITH  
THE USER GUIDE DATED AUGUST 28,1981. PLEASE CONSULT THE  
USER GUIDE DATED FEBRUARY 1,1983 FOR GUIDANCE.

THE FOLLOWING DATA IS REQUESTED OF THE USER AT THE  
TERMINAL SO DIFFERENT CALCULATIONS CAN BE MADE USING  
THE SAME DATA FILE

WHAT IS THE PROGRESS PAYMENT LAG TIME IN DAYS? (TIME  
FROM WHEN STATEMENT IS ISSUED UNTIL PAYMENT IS REC-  
EIVED?)

?  
>5

WHAT IS THE DELIVERY PAYMENT LAG TIME IN DAYS? (TIME  
FROM WHEN DELIVERY IS MADE UNTIL PAYMENT IS RECEIVED.)

?  
>20

WOULD YOU LIKE TO SPECIFY A CERTAIN PROGRESS PAYMENT  
RATE FOR THESE CALCULATIONS? A NO ANSWER IMPLIES YOU  
WANT THE COMPUTER TO CALCULATE THE FLEXIBLE PROGRESS  
RATE USING THE CONTRACTOR INVESTMENT RATIO. (YES / NO)  
>NO

AFTER HOW MANY DAYS WOULD YOU LIKE THE ALTERNATE  
LIQUIDATION RATE USED IN CALCULATIONS? VALUE MUST BE  
GREATER THAN OR EQUAL TO 365.

?  
>365

WOULD YOU LIKE TO SEE A TRANSACTION REPORT? (YES / NO)  
>NO

WOULD YOU LIKE TO SEE A RECAP OF THE COST REPORT DATA? (YES / NO)  
>NO

FIGURE 1

DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL  
VERSION DATED FEBRUARY 1, 1983  
ANALYTICAL REPORT

PROJECT TITLE CASH EXAMPLE WITH PROJECTED DATA

```
*****
CONTRACT PRICE          7700.00
TOTAL CONTRACTOR COST   7000.00
PROFIT PAID TO CONTRACTOR 700.00
START UP COST OF CONTRACT 0.00
CONTRACT START DATE     820101
DATA FILE PROFIT PERCENTAGE 10.00
COMPUTED PROFIT PERCENTAGE 10.00
*****
```

\*\*\* PROGRESS PAYMENT RATE DATA\*\*\*

```
*****
PROGRESS PAYMENT RATE    * 98.00 *
                        *      *
ORDINARY LIQUIDATION RATE * 98.00 *
                        *      *
ALTERNATE LIQUIDATION RATE * 89.09 *
                        *      *
CONTRACTOR INVESTMENT RATIO * 5.75 *
*****
```

\*\*\*\*\*  
\*\*\*\*\*

```

                                COST ELEMENT DATA
#  NAME          UNPAID COST  ZPAID  PAID COST  ELGIBILITY
                   DAYS FLOAT    DAYS FLOAT  PROGRESS PAYMENT
*****
1  DIRECT LABOR                -2      INCURRED COST
2  OVERHEAD          15                INCURRED COST
3  SUBCONT PROG PAY    20      00.0    -2      SUBCONTRACT PROG PAY
4  MATERIAL           35      00.0    -2      PAID COST
5  PURCHASED PARTS     20      90.0    -1      PAID COST
6  INTERDIVISIONAL     15      90.0    -4      PAID COST
*****
PROGRESS PAYMENT LAG TIME IN DAYS      5
DELIVERY PAYMENT LAG TIME IN DAYS      20
DATE OF FINAL DELIVERY                830525
```

```
PROGRESS PAYMENT RATE COMPUTED BY PROGRAM    98.00
*****
```

DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL  
VERSION DATED FEBRUARY 1, 1983  
ANALYTICAL REPORT

PROJECT TITLE CASH EXAMPLE WITH PROJECTED DATA

```
*****
CONTRACT PRICE          7700.00
TOTAL CONTRACTOR COST   7000.00
PROFIT PAID TO CONTRACTOR 700.00
START UP COST OF CONTRACT 0.00
CONTRACT START DATE     820101
DATA FILE PROFIT PERCENTAGE 10.00
COMPUTED PROFIT PERCENTAGE 10.00
*****
```

\*\*\* PROGRESS PAYMENT RATE DATA\*\*\*

```
*****
PROGRESS PAYMENT RATE    * 98.00 *
                          *      *
ORDINARY LIQUIDATION RATE * 98.00 *
                          *      *
ALTERNATE LIQUIDATION RATE * 89.09 *
                          *      *
CONTRACTOR INVESTMENT RATIO * 5.42 *
*****
```

\*\*\*\*\*  
\*\*\*\*\*

```

                                COST ELEMENT DATA
#  NAME                UNPAID COST  ZPAID  PAID COST  ELGIBILITY
                        DAYS FLOAT    DAYS FLOAT  PROGRESS PAYMENT
*****
1  DIRECT LABOR                15                -2    INCURRED COST
2  OVERHEAD                    15                -2    INCURRED COST
3  SUBCONT PROG PAY            20             80.0    -2    SUBCONTRACT PROG PAY
4  MATERIAL                    28             80.0    -2    PAID COST
5  PURCHASED PARTS             20             90.0    -1    PAID COST
6  INTERDIVISIONAL             15             90.0    -4    PAID COST
*****
PROGRESS PAYMENT LAG TIME IN DAYS        6
DELIVERY PAYMENT LAG TIME IN DAYS        20
DATE OF FINAL DELIVERY                   830525
```

```
PROGRESS PAYMENT RATE COMPUTED BY PROGRAM    98.00
*****
```

## 1. INTRODUCTION CONT'D

-----

The first block of information in Figure 2 is a combination of program calculations and facts contained in the input file. Total contractor cost is the sum of all cost elements contained in the file. Profit paid is total cost subtracted from contract price. Contract price is derived by adding all the deliverables. Profit paid and contractor cost are then used to calculate profit percentage. The calculated profit percentage is then compared to the percentage specified in the datafile. If the two percentages are not the same, an error message is printed and the program stops. Start up cost and contract start time are simply read in from the input file

Now the program calculates the progress payment rate, the alternate liquidation rate and the contractor investment ratio. Ordinary liquidation rate is always the same as the progress payment rate. These four numbers are the real meat of this report. The "what if" analysis is performed to see the effect on them.

The rest of the numbers shown in Figure 2 are simply being reported from the input file. You should verify that these numbers are what you intended. If not, the input file should be changed. Also included here are the values that you input to be used as the progress payment and delivery payment time lags.

To complete our initial look at the CASHII program let's turn to Figure 3. It shows the input file that was used to generate Figure 2.

820101,0.6,20.1

CASH EXAMPLE W11 PROJECTED DATA

DIRECT LABOR, INCUR, -2,0,0

SECTION 1

OVERHEAD, INCUR, 15,0,0

SUBCONT PROC PAY, SUBCT, 20, .6, -2

MATERIAL, PAIDC, 35, .8, -2

PURCHASED PARTS, PAIDC, 20, .9, -1

INTERDIVISIONAL, PAIDC, 15, .9, -4

SECTION 2

3,820130,0,0,0,100,0,0

3,820220,0,0,0,100,0,0

3,820330,150,50,0,0,0,0

4,820425,250

3,820430,150,50,100,100,100,100

3,820530,150,50,100,100,100,100

3,820630,150,50,100,100,100,100

3,820730,150,50,100,100,100,100

3,820830,150,50,100,100,100,100

4,820925,500

SECTION 3

3,820930,150,50,-100,300,100,100

3,821030,150,50,-100,300,100,100

3,821130,150,50,-100,300,100,100

4,821224,1000

3,821231,150,50,-100,300,100,0

4,830125,1000

3,830130,150,50,-100,300,0,0

4,830219,1000

3,830220,150,50,0,200,0,0

3,830330,150,50,0,0,0,100

3,830430,150,50,0,0,0,0

4,830525,3950

FIGURE 3

## 1. INTRODUCTION CONT'D

-----

### INPUT DATA FILE

The input data file is actually made up of three sections. A detailed discussion of the information contained in each is found in the next section of this document. For now, the first two lines contain information about the rest of the file and the title of the project. In this example, the next six lines contain information about the individual cost elements that will make up the contract expenses. (You may have up to 10 lines of cost elements but you have to specify how many you have.)

The rest of the file contains the transactions that take place over the life of the contract. A transaction type of 3 means a monthly cost statement (either actual or estimated), the number after the 3 is the date and the following six numbers are the actual (or estimated) costs for each of the cost elements specified above. A transaction type 4 is a planned delivery. The number after the 4 is the date and the next number is the value of the deliverable.

Now you know nearly everything necessary to use the CASH11 program. If your contract used the six cost elements illustrated in this example, the only thing we haven't discussed is the rest of the numbers on the cost element line. Each of them is discussed in the next section.

## 1. INTRODUCTION CONT'D

---

### HOW TO USE THE REST OF THIS BOOK

---

If you have read this far you have a good fundamental understanding of how CASHII works. The next step depends on how you will be using it. If you plan to make extensive use of the system you should study sections 2, 3 and the appropriate appendix. If you feel you have already seen enough, you may be able to squeak by with section 2 and the appropriate appendix. Section 2 is needed by everyone since it describes in detail what the items in the data file are. You may find that you can skip much of it simply by referring to section 4, the short guide to syntax for specific questions or rules and limitations.

Section 4 will probably prove useful to browse through. However, it is not intended to be read from start to finish. Its function is that of a dictionary for the CASHII system. The last part of it addresses error messages. If when you execute CASHII, unexpected conditions occur you may be able to obtain help by looking there.

Finally, the appendix which is most relevant will depend on exactly how you are using CASHII. The program itself is maintained on two commercial computer service vendors. One appendix demonstrates how CASHII works on each. CASHII is also available for your own in-house computer. (CASHII is written in FORTRAN) If that's where you are using it most of this book is useful. However, the appendices of sample runs may not be of any value.

## 2. CASHII BASICS

### GETTING STARTED

First you must decide where you will be using CASHII. One important consideration is who will be using it (actually entering data and executing the program). If it is someone with no previous computer experience you should look for a "user friendly" system. That means a computer system which is as simple as possible to operate and spends as little time as possible with computerese. You should also plan on at least some introductory training in the particular computer system and how it works. (This book should be all you need on the application, if it's not, there is an on-line teaching tool which should help.)

If the typical users will be computer professionals none of the above is of particular concern. You will probably want to get your own copy of the program and run it on your own in-house computer. Some regular system should be set up to insure that you are kept up-to-date with the latest version. If you elect to use your own in-house system, skip to Creating the Data File. The rest of this discussion is for those who will be running the program on an outside vendor and who have no particular computer experience.

The next step is access to one of the commercial vendors. DOD does not provide this access. You will have to make arrangements directly with the vendor to become a customer. It's important that you inform the vendor that you wish to use the CASHII system. They may have some special arrangements that must be made.

GOVERNMENT Note for government users: Your access to CASHII is  
USERS NOTE through Copper Impact on Boeing Computer Services only.



## 2. CASHII BASICS CONT'D

### GETTING STARTED

#### SELECTING A TERMINAL

After picking your vendor, you will want to discuss with them the type of terminal device necessary to access their computer. The key consideration is how much time you will spend using CASHII. If you expect to spend a substantial amount of time using CASHII then a higher speed terminal is a good investment. For your purposes terminals operate at 30 or 120 characters per second (CPS). A full set of output for the sample problem presented in section 2 takes about 10 minutes to print at 30 CPS. If you anticipate doing one similar run per week then 30 CPS should be adequate. For one a day or for contracts that have much more data than that illustrated 120 CPS is preferable and possibly even more economical. Your vendor can help you determine more accurately what volume will make the higher speed devices the most economical.

Of course, if you already have a terminal chances are good that it will be compatible with the computer you select. If you elect to attend an introductory training session on the vendors computer service, you should ask that specific discussion be devoted to your terminal and how it works.

That should be all it takes to get started. Now you are ready to enter data. That's the next topic.

## 2. CASHII BASICS CONT'D

### CREATING A DATA FILE

We have already had a brief discussion of the data file. You know what it looks like and that it is divided into three sections. Now we will discuss each of the sections in sufficient detail to enable you to write down your data in the proper format. Note that when you are preparing to run the CASHII program, you must have entered (and saved) the appropriate data into the data file. That's the first step. Then CASHII reads the information from the file. The actual steps of entering the data file are shown in the appropriate appendix. You will see as we go along that it makes sense to write the data on paper in the proper format first.

The first line in our example is:

820101,0,6,22,.1

A	A	A	A	A
				Planned profit percent as a decimal.
				This example plans on 10% profit.
				The number of transaction lines in this
				data file (more on this in the discussion
				of section 3).
				The number of cost elements associated with
				this contract Maximum of 10 allowed.
				Start up costs.
				Start up date (year, month, day format)

The second line is:

### CASHII EXAMPLE WITH PROJECTED DATA

This is simply the title that we wish to print at the top of each of the reports. You can have any title you wish up to 50 characters long.

## 2. CASHII BASICS CONT'D

### CREATING A DATA FILE

That completes the first section of the file. The second section is where you specify each of the cost elements which will be involved with the contract. Each cost element requires that you specify the type of cost (as specified in the progress payment clause of the contract) followed by three items of numeric data.

The type of cost must be INCUR, PAIDC or SUBCT.

A typical cost element line (as shown in our earlier example) is:

DIRECT LABOR, INCUR, -2, 0, 0

↑                    ↑                    ↑                    ↑  
Title (maximum of 20 characters)    Type of cost    Days float    Dummy elements for INCUR must be 0.

#### COST ELEMENT TYPES

INCUR means that expense associated with this cost element are eligible for progress payment as they are incurred.

PAIDC means that expenses are eligible for progress payments only as they are actually paid.

SUBCT means that this cost element represents progress payments paid to a subcontractor and are eligible for 100% reimbursement to the prime contractor.

Important note for large business firms: Some contract costs such as material or purchased parts costs incurred are not eligible for progress payments until actually paid. Therefore, these elements must be coded as PAIDC.

## 2. CASHII BASICS CONT'D

### CREATING A DATA FILE

FLOAT is the next item on the line. Slightly different considerations are necessary depending on whether the cost element is INCUR, PAIDC or SUBCT. Following the example, let's look at INCUR first.

FLOAT is the number of days between the cost statement date and the time the cost is paid on a cash basis. Bank lag times must be included in float. Both positive and negative numbers may be entered.

Float for labor should be based on a weighted average over the cost accounting period. For example, if the cost accounting period is 4 weeks and the contractor pays every 2 weeks, and it takes an average of 8 days from the time the payroll is closed until the payroll checks clear the contractor's bank account, the average float for the 4 week period would be 1 day, computed as follows:

		PAYROLL PERIOD	
		FIRST	SECOND
FLOAT EXAMPLE	DAY PAYROLL CLOSES	14	28
	PLUS: PAYMENT LAG	8	8
	PAYROLL PAYMENT DATE	22	36
	LESS: DAY OF COST		
	STATEMENT	28	28
	FLOAT	-6	+8
	TIMES: PAYROLL WEIGHT*	.5	.5
	WEIGHTED FLOAT	-3	+4

AVERAGE FLOAT  $(-3+4) = 1$  DAY

\*Expected dollar amount is the same for each payroll period so they are equally weighted. The average float is then the sum of the two individual float amounts.

## 2. CASHII BASICS CONT'D

### CREATING A DATA FILE

The next cost element is also type INCUR. It is:

OVERHEAD, INCUR, 15, 0, 0

This follows exactly the same format as we just discussed. However, calculating float for overhead items is intricate enough that we should step through a specific example.

Overhead will include things like indirect labor, vacation, depreciation, materials and supplies. These things will typically have very different float times so it is necessary to calculate a weighted average. The example below illustrates the calculation.

	ELEMENT OF EXPENSE	AMOUNT	PERCENTAGE	FLOAT	WEIGHTED
			OF TOTAL	DAYS	FLOAT DAYS
FLOAT EXAMPLE	Indirect labor	\$ 1,700	17.0%	-2	-.34
	Payroll taxes	1,500	15.0	-6	-.90
	Holidays	600	6.0	29	1.74
	Sick Pay	400	4.0	30	1.20
	Accrued vacation	1,000	10.0	60	6.00
	Retirement	1,500	15.0	42	6.30
	Insurance	500	5.0	-15	-.75
	Depreciation-Plt&Equip	1,200	12.0	0	.00
	Travel	300	3.0	9	.27
	Materials & Supplies	600	6.0	20	1.20
	Miscellaneous	700	7.0	4	.28
	TOTAL	\$10,000	100.0%		15.00

## 2. CASHII BASIC CONT'D

---

### CREATING A DATA FILE

---

Note the significant number of float days associated with vacation. Vacation expense is accrued long before it is actually paid. Since we are calculating a weighted average the overall effect is consistent with its importance to total expenses. Also notice that depreciation float is shown as zero. Facilities capital cost of money should also be treated as having no float.

Weighted float days is percent of total times float days. The weighted float days are then added and the total used as the float time for this cost element.

Since only ten cost elements are allowed, it makes sense to combine similar cost elements in this manner whenever necessary.

The next line in our data file illustrates a SUBCT expense. The format for SUBCT and PAIDC cost elements is the same so we will discuss them together.

## PAIDC and SUBCT COST ELEMENTS

---

Let's start by showing the example we are working with:

SUBCONT PROG PAY, SUBCT, 20, .8,-2

↑	↑	↑	↑	↑
				Float for paid costs.
				Percent of cost paid in
				current cost accounting period.
				Float for unpaid costs.
				Type of cost element.
				Cost element.

We can deal with the two float numbers with an example (which will be equally applicable to PAIDC cost elements) so let's discuss the percent of cost paid first. This number is the portion of costs incurred during the accounting period for which the contractor has already issued checks. This cost is eligible for progress payments. Note, the number must be expressed as a decimal. In this case, the contractor has paid 80% of the expenses during the accounting period.

The first float item is specified as float for unpaid costs. This is the number of days after the cost statement is made that the expense is paid. The second float item is float for paid costs. This is the number of days before the cost statement is made that this category of expenses is paid. Calculations for both these numbers can be illustrated in one example.

## 2. CASHII BASICS CONT'D

### CREATING A DATA FILE

The contractor made a review of invoices received during April to develop these data. Material costs are recorded on an accrual basis. To keep the example simple, assume that invoices are paid 20 days after receipt. Invoices selected for this computation should cover a period of time sufficient to represent the contractor's normal dollar volume and payment cycle for such purchases. Cost elements involving an insignificant amount in relation to the contractor's normal volume need not be analyzed.

#### SAMPLE MONTH - APRIL

DATE INVOICE RECEIVED (1)	INVOICE AMOUNT (2)	CHECK WRITTEN (3)	BANK FLOAT (4)	TOTAL		DAY DOLLARS	
				DAYS	DAYS FROM	(DAYS FROM START OF APR)	
				START OF	TIMES INVOICE	AMOUNT)	
				APRIL	APRIL	MAY	
				(5=3+4)	(6=2*5)		
APR 1	\$ 100	APR 21	3	24	\$ 2,400	}	paid costs
2	150	22	3	25	3,750		
3	200	23	3	26	5,200		
7	175	27	3	30	5,250		
8	125	28	3	31	3,875		
9	50	29	3	32	1,600	}	un-paid costs
21	25	MAY 11	3	44	\$1,100		
23	50	13	3	46	2,300		
29	100	19	3	52	5,200		
30	25	20	3	53	1,325		

TOTAL \$1,000

(APR TOTAL \$800.

MAY TOTAL \$200.)	TOTAL DAY DOLLARS	\$22,075	\$9,925
Divided by: Total amount paid each month		800	200
Weighted average payment days		28	50
Less: Number of days in April		<u>-30</u>	<u>-30</u>
Float Days		-2	-20

#### LIMITATION ON FLOAT FOR PAIDC & SUBCT

In view of the requirements of the Prompt Payment Act, to encourage prompt payment of subcontractors, the float on the unpaid portion of PAIDC or SUBCT categories of cost shall not exceed 28 days, notwithstanding any data the contractor may have which would indicate a historically longer float period.



## 2. CASHII BASICS CONT'D

### CREATING A DATA FILE

Now to find the weighted average float for paid expenses (those paid for in April) simply divide the total day dollars by the April total invoices paid amount. So --

$$\frac{22075}{800} = 27.5 = 28 \text{ days}$$

Now subtract the number of days in April 28-30 = -2 and arrive at -2 as the float for paid expenses. In other words you paid expenses on average 2 days before the cost statement was issued.

The calculation for unpaid expenses works the same way:

$$\frac{9925}{200} = 49.6 = 50 \text{ days}$$

50-30 days in April = 20 days float.

These expenses are not paid until 20 days after the cost statement is issued.

It is important that the items used in this analysis be similar. For instance, if subcontractor progress payments are made much quicker than other payments, a separate analysis should be performed for each.

You can also see that the April total of \$800 represents 80% of the total invoices received. That's where we got the percent of cost paid number.

PAIDC cost elements can be viewed in exactly the same way, so a separate example will not be given.

The rest of the lines in this part of the data file follow the formats just discussed. Now let's move on to the third section.

## 2. CASHII BASICS CONT'D

### CREATING THE DATA FILE

TRANSACTIONS - You can see from looking at the example that several different formats are followed in the third section of the data file. The format varies according to the type of transaction. The transactions are identified by the numbers 1, 2, 3, 4, and 5. Here is what each means:

- 1) Actual progress payment received. These will be entered into the input data file as appropriate to reflect receipt of payment. Contains three items, date, amount, progress payment rate.
- 2) Planned progress payment. These transactions show up in the transaction report. They are calculated by and used only by the CASHII program.
- 3) Monthly cost statement (actual or estimated). Usually make up the bulk of the transactions entered in the data file. Contains date plus one entry for each cost element, zero if no value.
- 4) Planned delivery. Contains date and value.
- 5) Actual delivery. Contains date, value, liquidation rate.

All transactions must be entered in order by date. The total number of transaction lines must be equal to the number of transactions specified in the first line of the data file. You use transaction types 1, 3, 4 and 5. The program uses transaction type 2. The total number of transactions allowed in any one run of the program is 325. Since the type 2 transactions used by the program count toward this limit your limit is somewhat less. You can see if you are within the limit by trying the calculation:

$$2 * (\text{\#of type 3's}) + (\text{\#of 1's} + \text{\#of 4's} + \text{\#of 5's})$$

If the answer you get is less than 325 you should be OK.

Now let's look at each type of transaction.

## 2. CASHII BASICS CONT'D

### CREATING THE DATA FILE

Each line in the third section of the data file must specify which type of transaction it is. In our example, the next line is:

3,820130,0,0,0,100,0,0

↑     ↑                     ↑  
transaction type     date of transaction     cost element data

The transaction illustrated is a type 3 or Monthly cost statement transaction. This portion of the file must be in order by the transaction dates. So, the next line, no matter what type of transaction it represents must have a date greater than January 30, 1982. Transactions of type 3 may be either actual or estimated expenses.

The six data items following the date correspond to the six cost elements specified in section 2 of the file. Since we specified a 6 on the first line of the file, each type 3 line must have six cost items or the program will not work. Can you tell which cost element incurred the \$100 expense shown above?

Right, it's the fourth one or in our example, Material expense. We are saying here that for the transaction dated 820130 only material expenses were planned or actually incurred. If labor expense had been incurred, it would replace the first zero after the date. If interdivisional expenses were incurred, it would replace the last zero on the line and so forth. Note: that commas are used to separate the individual items of data on the line. Therefore, the system will be confused if you insert commas within the numbers. Remember, as shown in our example the number one thousand should be entered as 1000 NOT as 1,000.

COMMAS  
REQUIRED

## 2. CASHII BASICS CONT'D

### CREATING THE DATA FILE

Our example continues with two more type 3 transactions. You should agree that the second of them shows a date of March 30, 1982 and specifies \$150 direct labor expense and \$50 overhead expense. If you don't agree, study the preceeding page of explanation on the layout of this line.

The next line is a transaction type 4 or planned delivery transaction. In our example it looks like this:

```
4,820425,250
  ^      ^      ^
  |      |      | dollar value of planned delivery
  |      |      | date of planned delivery
  |      |      | transaction type
```

Transaction type 4's require only two items on the line after the transaction type.

So far we have discussed transaction types 2 (used by the program only) 3 (Monthly Cost Statements) and 4 (Planned Delivery). That leaves 1 and 5. Both of them represent Actuals -- a 1 is an actual progress payment, and a 5 is an actual delivery payment.

At the beginning of a contract, your file may well look like the example we have used so far. That is, agreement was reached as part of the original negotiations to request flexible progress payments. Sometimes, however, a contract is already underway before the decision is made to request flexible payments. In this situation the type 1 transaction is used to reflect any actual progress payments received.

Let's modify our sample data file to reflect some actual payments received.

## 2. CASHII BASICS CONT'D

### CREATING THE DATA FILE

more transaction data

820101,0,6,25,.1  
 cash example with actual and projected data ← new title just for  
 direct labor,incur,-2,0,0 illustration  
 overhead,incur,15,0,0 purposes  
 subcont prog pay,subct,20,.9,-2  
 material,paidc,28,.8,-2  
 purchased parts,paidc,20,.9,-1  
 interdivisional,paidc,15,.9,-4  
 3,820130,0,0,0,100,0,0  
 1,820204,0,.9 ←  
 3,820228,0,0,0,100,0,0  
 1,820304,90,.9 ← actual progress payments received  
 3,820330,150,50,0,0,0,0  
 1,820404,270,.9 ←  
 4,820425,250  
 3,820430,150,50,100,100,100,100  
 3,820530,150,50,100,100,100,100  
 3,820630,150,50,100,100,100,100  
 3,820730,150,50,100,100,100,100  
 3,820830,150,50,100,100,100,100  
 4,820925,500  
 3,820930,150,50,-100,300,100,100  
 3,821030,150,50,-100,300,100,100  
 3,821130,150,50,-100,300,100,100  
 4,821224,1000  
 3,830130,150,50,-100,300,0,0  
 4,830219,1000  
 3,830228,150,50,0,200,0,0  
 3,830330,150,50,0,0,0,100  
 3,830430,150,50,0,0,0,0  
 4,830525,3950

## 2. CASHII BASICS CONT'D

### CREATING THE DATA FILE

The changes between this example and the ones we have worked with so far are flagged. You will see that we added three transactions so the item in line one that tells the program how many transactions to expect was also changed to 25 from 22. The three new transactions are type 1 transactions. The type 1 transaction (or actual progress payment) format is:

```
1,820204,0,.9
  ^   ^   ^   ^
  |   |   |   |
  |   |   |   | Progress payment rate (90% here)
  |   |   |   | Progress payment actually received (see below
  |   |   |   | for why this one is zero)
  |   |   |   | Date payment received
  |   |   |   | Transaction type
```

Type 1 transactions show receipt of actual progress payments.

In this case, since the payment rate is 90%, you can tell the example is for a large business concern. These actual progress payments are always at the uniform or standard progress payment rate. And they occurred before flexible payment rate was requested.

This payment was 0 simply because the payment value would have been so small it was not requested by the contractor. Since the contract illustrated was already in progress and there must be a one to one match between cost statements and actual (or planned) progress payments, this transaction was inserted.

## CREATING THE DATA FILE

```

820101,0,6,25,.1
cash example with actual and projected data
direct labor,incur,-2,0,0
overhead,incur,15,0,0
subcont prog pay,subct,30,.9,-3
material,paidc,28,.8,-2
purchased parts,paidc,20,.9,-1
interdivisional,paidc,15,.9,-4
3,820130,0,0,0,100,0,0
1,820204,0,.9
3,820228,0,0,0,100,0,0
1,820304,90,.9
3,820330,150,50,0,0,0,0
1,820404,270,.9
5,820425,250,.9 ←————— actual delivery payment received
3,820430,150,50,100,100,100,100
3,820530,150,50,100,100,100,100

```

Comparing to the earlier example you will see that the transaction 5 replaced a type 4 transaction. The delivery has actually taken place and payment was received. Since the 5 replaced a 4, the number of transactions in the file remains the same and no change is required in the first line. The format for a type 5 transaction, actual delivery payment received is:

5,820425,250,.9

A        A        A        A

|        |        |        |

Liquidation rate (90% here)

Value of delivery

Date delivery made

Transaction type

## 2. CASHII BASICS CONT'D

### CREATING THE DATA FILE

#### LIQUIDATION RATES

The liquidation rate is part of a delivery transaction. There are two liquidation rates, ordinary and alternate. For further discussion see DAR Appendix E Part 5.

We didn't discuss a type 2 transaction's format in the data file. That's because type 2 or planned progress payment transactions are used only by the program. You are not allowed to have any of them in your data file.

If you specify when you run CASHII that you want to see a transaction report, CASHII will print type 2 transactions in the appropriate spots. The transaction report is your best tool for seeing the planned cash flow and timing of the contract. It is discussed in section 3, CASHII output.

Now that we have completed our input file, let's talk about modifying it.



## 2. CASHII BASICS CONT'D

---

### CHANGING/CORRECTING YOUR DATAFILE

---

Simple, Type it over.

Actually, you will probably want to change a specific item on one or a few lines rather than the whole file. In that case, it probably is simplest to just type the lines in question over again.

One of the strengths of computers is the ability to let the machine do the work for you. There is undoubtedly a way to instruct whichever computer you are using to change some specific data without retyping the entire line. There are some examples of doing this in the appendix of sample runs. If you don't happen to be using one of those vendors, try looking in a reference manual on System Commands or Editing Commands or try typing the line over.

Now you are ready to execute CASHII.

## 2. CASHII BASICS CONT'D

### EXECUTING CASHII

Once again, the precise mechanism for executing CASHII depends on the computer you are using. Some specific examples are given in the appendix. However, in every case you must go through the following steps:

Access and sign on to the computer of your choice.

Enter the data required into a data file and save it.

Execute the program.

Sign off the computer.

The balance of this section is devoted to answering the questions asked by CASHII.

Once you have begun execution of CASHII, the first question will be "enter input file name". Supply the name of the file that contains your data in response to this question.

Next the program asks you to

Enter the progress payment lag time.

and

What is the delivery payment lag time?

## 2. CASHII BASICS CONT'D

### EXECUTING CASHII

LIMIT ON  
LAG TIME

These two numbers are very significant in terms of the overall analysis. Progress payment lag tends to be the more significant. Two rules must be followed or CASHII will not process your data. Progress payment lag time may not exceed 56 days. Delivery payment lag time may not exceed 60 days.

Longer lag times tend to increase the contractor investment ratio and if significant enough, increase the progress payment rate.

The next question enables you to specify a particular progress payment rate. If you answer yes, CASHII asks you to enter the rate. Answering no allows the program to compute the rate.

The next question is: After how many days would you like the alternate liquidation rate used?

The alternate liquidation rate cannot be used until at least 365 days into the contract.

The next question is: Would you like to see a transaction report?

REPORT  
WIDTH

Answering yes provides the output discussed in the next section. Note that the report is 125 characters wide. The transaction report is useful for determining the actual cash flow of the contract. If you are interested only in the progress payment rate and the contractors investment ratio, it can be skipped.

The final question is: Would you like to see a recap of the cost report data?

Answering yes provides the output discussed in the next section. This report is 132 characters wide. The recap of cost statements can be very useful in detecting a typing mistake on the input data file. It simply formats nicely and prints all the type 3 transaction data from the file, with totals. You may routinely answer no unless you expect a problem in the data.

## 2. CASHII BASICS CONT'D

-----

### EXECUTING CASHII

-----

Now CASHII produces the output you requested and the Analytical Report. On completion of all the output, one final question is asked:

Would you like to run the model again using the same datafile but changing the terminal input?

Answering yes starts the run over again at the question "Enter progress payment lag time". Answering no stops the program.

If you have another data file that you wish to process, you should answer no, let the program stop and then execute it again.

See the appendix for complete annotated examples of executing CASHII.

Now we are ready to discuss the output in detail. That's the subject of the next section.

### 3. CASHII OUTPUT

#### AVAILABLE REPORTS

CASHII offers to produce three reports each time it is executed. One of them, the Analytical Report, is always produced. An example of it was shown in section 1, A Sample CASHII Session. It answers the question; What will the flexible progress payment percentage be? The other two reports are both optional and are produced only if you request them. They are the Transaction Report and the Recap Cost Statement Report. Both of them round to the nearest whole number.

Let's address the reports in the same order in which they are produced.

#### TRANSACTION REPORT

If you answered yes for the transaction report, that will be the first one generated. Figure 4 shows what it looks like:

DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL  
VERSION DATED FEBRUARY 1, 1983  
TRANSACTION REPORT

CASH EXAMPLE WITH PROJECTED DATA

DATE / TRANSACTION CODE	1 TOTAL INCURRED COST	2 FLUAT	3 PAID INCURRED COST	4 FROM PAY REC'D	5 PAID AT DELIVERY	6 TOTAL COST REIMBURS (4+5)	7 PAID AT DELIVERY	8 TOTAL REVENUE (6+7)	9 NET IM PAID (8-6)	10 NET TO REVENUE PAYMENTS
(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)
820130 / 3	100	20	80	0	0	0	0	0	80	0
820205 / 2	100	20	80	78	0	78	0	78	2	78
820228 / 3	200	20	180	78	0	78	0	78	102	78
820306 / 2	200	20	180	176	0	176	0	176	4	176
820330 / 3	400	50	350	176	0	176	0	176	174	176
820405 / 2	400	50	350	392	0	392	0	392	-42	392
820430 / 3	1000	110	890	392	0	392	0	392	498	392
820506 / 2	1000	110	890	923	0	923	0	923	-38	923
820515 / 4	1000	50	950	923	5	927	0	928	23	678
820530 / 3	1600	110	1490	923	5	927	0	928	563	678
820605 / 2	1600	110	1490	1513	5	1517	0	1518	-27	1268
820630 / 3	2200	110	2090	1513	5	1517	0	1518	573	1268
820706 / 2	2200	110	2090	2103	5	2107	0	2108	-17	1958
820730 / 3	2800	110	2690	2103	5	2107	0	2108	583	1958
820805 / 2	2800	110	2690	2693	5	2697	0	2698	-7	2448
820830 / 3	3400	110	3290	2693	5	2697	0	2698	593	2448
820905 / 2	3400	110	3290	3283	5	3287	0	3288	3	3038
821006 / 3	4000	110	3890	3283	5	3287	0	3288	603	3038
821015 / 4	4000	50	3950	3870	5	3874	0	3875	16	3625
821030 / 3	4600	110	4490	3870	14	3883	1	3885	607	3155
821105 / 2	4600	110	4490	4456	14	4469	1	4471	21	3721
821130 / 3	5100	110	5090	4456	14	4469	1	4471	621	3721
821206 / 2	5200	110	5090	5042	14	5055	1	5057	35	4307
821231 / 3	5700	100	5600	5042	14	5055	1	5057	545	4307
830106 / 2	5700	100	5600	5539	14	5553	1	5554	47	4804
830113 / 4	5700	100	5600	5539	32	5571	159	5730	489	3960
830130 / 3	6100	90	6010	5539	32	5571	159	5730	39	4380
830205 / 2	6100	40	6060	5539	50	5589	250	5839	71	3488
830228 / 3	6500	90	6410	5539	50	5589	250	5839	421	3488
830306 / 2	6500	90	6410	6381	50	6431	250	6681	29	2990
830311 / 4	6500	60	6440	6381	48	6429	241	6670	341	2990
830330 / 3	6800	60	6740	6381	68	6449	341	6790	18	3313
830405 / 2	6800	50	6750	6754	68	6822	341	7164	228	3313
830506 / 3	7000	50	6950	6860	68	6918	341	7259	22	3519
830614 / 4	7000	0	7000	6860	140	7000	700	7700	0	0

FIGURE 4

### 3. CASHII OUTPUT CONT'D

-----

Working across the report from left to right you see that each transaction has generated 10 columns of information. This includes the type 2 transactions, Planned Progress payments which are inserted by CASHII. Let's look at those columns which may not be self-explanatory:

Column 3 is calculated as column 1 minus column 2.

Column 5 involves several calculations so it is best explained using our example. Note, that the delivery amount of the first planned delivery is \$250, the ordinary liquidation rate is .98 and the profit rate is .10. The calculations are:

$$250 \times .98 = 245 = \text{liquidation amount}$$

$$250 / (1 + .10) = 227.3 = \text{cost in this delivery}$$

$$227.3 \times .98 = 222.7 = \text{cost previously reimbursed as progress payments}$$

$$227.3 - 222.7 = 4.6 \text{ cost reimbursed at delivery} \\ (\text{rounds to 5}) \text{ displayed in column 5.}$$

Column 7 also involves some calculating using some of the same numbers as column 5:

$$250 - 245 = 5 = \text{net payment at delivery}$$

$$5 - 4.6 = .4 = \text{profit paid at delivery (rounds to 0)} \\ \text{displayed in column 7.}$$

Column 10 is calculated by subtracting the liquidation amount (calculated in column 5) from cumulative progress payments.

### 3. CASHII OUTPUT CONT'D

#### ANALYTICAL REPORT

The analytical report is always produced. It contains the key items of information that CASHII is all about. Namely, the progress payment rate and the contractor investment ratio. Recall from our earlier discussion that the progress payment rate may range up to 100% while the contractor investment ratio is held at approximately 5%. Details on the method of calculating the contractor investment percentage are found in section 4, short guide to syntax.

Figure 5 is a repeat of the sample Analytical Report shown earlier:



DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL  
VERSION DATED FEBRUARY 1, 1983  
ANALYTICAL REPORT

PROJECT TITLE CASH EXAMPLE WITH PROJECTED DATA

```
*****
CONTRACT PRICE          7700.00
TOTAL CONTRACTOR COST   7000.00
PROFIT PAID TO CONTRACTOR 700.00
START UP COST OF CONTRACT 0.00
CONTRACT START DATE     820101
DATA FILE PROFIT PERCENTAGE 10.00
COMPUTED PROFIT PERCENTAGE 10.00
*****
```

\*\*\* PROGRESS PAYMENT RATE DATA \*\*\*

```
*****
PROGRESS PAYMENT RATE    * 98.00 *
ORDINARY LIQUIDATION RATE * 98.00 *
ALTERNATE LIQUIDATION RATE * 89.09 *
CONTRACTOR INVESTMENT RATIO * 5.42 *
*****
```

```
*****
*****
```

#	NAME	COST ELEMENT DATA		ELGIBILITY
		UNPAID COST DAYS FLOAT	PAID COST DAYS FLOAT	
1	DIRECT LABOR		-2	INCURRED COST
2	OVERHEAD	15		INCURRED COST
3	SUBCONT PROG PAY	20	80.0	SUBCONTRACT PROG PAY
4	MATERIAL	28	80.0	PAID COST
5	PURCHASED PARTS	20	90.0	PAID COST
6	INTERDIVISIONAL	15	90.0	PAID COST
*****				
	PROGRESS PAYMENT LAG TIME IN DAYS	6		
	DELIVERY PAYMENT LAG TIME IN DAYS	20		
	DATE OF FINAL DELIVERY	830525		

```
*****
PROGRESS PAYMENT RATE COMPUTED BY PROGRAM    98.00
*****
```

FIGURE 5

### 3. CASHII OUTPUT CONT'D

#### ANALYTICAL REPORT

The first block of information consists of a summary of data from your data file and results of CASHII's calculation of profit percentage. The calculated profit percentage and the percent specified in the first line of your data file must be the same or the program will stop.

Contract price is the sum of all the deliverables in the data file. Total contractor cost is the sum of all the cost elements in the datafile. The difference between those two numbers is the profit. Start up cost and start date are simply read in from the data file.

Next is a block showing the result of the calculations which have taken place. In this example the progress payment rate was calculated to be 98%. The ordinary liquidation rate is always the same as the progress payment rate. Liquidation rate is the portion of any deliverable amount which is assumed to have already been reimbursed. The alternate liquidation rate is calculated by CASHII and can be used, at your option, after 365 days of the contract. Finally, the contractor investment ratio is calculated. This number should not be less than 5%.

The next block of data on the Analytical Report is a formatted printout of your cost element data from the input data file. You should verify that these numbers are what you expected. If not, it will be necessary to change the data file.

Finally, the Analytical Report shows you how you answered the two lag time questions asked at the terminal and the date of final delivery as specified in the data file.

### 3. CASHII OUTPUT CONT'D

#### RECAP OF COST STATEMENT REPORT

The Recap Report is produced last, if you requested it. Here's a sample of what it looks like:

PAGE 4

DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL  
VERSION DATED FEBRUARY 1, 1983  
RECAP OF COST STATEMENT  
NOTE: TOTAL FOR FIRST COST STATEMENT INCLUDES START UP COST

#### CASH EXAMPLE WITH PROJECTED DATA

DATE	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	TOTAL
820100	0	0	0	100	0	0	0	0	0	0	100
820200	0	0	0	100	0	0	0	0	0	0	100
820300	150	50	0	0	0	0	0	0	0	0	200
820400	150	50	100	100	100	100	0	0	0	0	600
820500	150	50	100	100	100	100	0	0	0	0	600
820600	150	50	100	100	100	100	0	0	0	0	600
820700	150	50	100	100	100	100	0	0	0	0	600
820800	150	50	100	100	100	100	0	0	0	0	600
820900	150	50	-100	300	100	100	0	0	0	0	600
821000	150	50	-100	300	100	100	0	0	0	0	600
821100	150	50	-100	300	100	100	0	0	0	0	600
821200	150	50	-100	300	100	0	0	0	0	0	500
830100	150	50	-100	300	0	0	0	0	0	0	400
830200	150	50	0	200	0	0	0	0	0	0	400
830300	150	50	0	0	0	100	0	0	0	0	300
830400	150	50	0	0	0	0	0	0	0	0	200
TOTAL	2100	700	0	2400	900	900	0	0	0	0	7000

Careful study will show that this report pulls all the transaction type 3's from your data file, formats them and prints them for your inspection.

As mentioned previously, this is a good way to check the validity of your input data. If CASHII is giving unexpected results, check here to see that the computer actually has the data you think it does.

One other point of interest is the negative numbers in cost element #3, Sub-contractor Progress Payments. These simply indicate liquidation of earlier entries. The total for subcontractor cost elements must add to zero over the life of the contract.

#### 4. CASHII TERMS, DEFINITIONS AND SYNTAX

---

CASHII is a very simple program to run. The questions it asks and the error messages it provides are as clear as possible and provide as much information as possible.

Syntax refers to exactly how you must provide information so that it will be understood by the program. For instance, items in the datafile must be separated by commas and percentages must be expressed as a decimal. In fact, those are the two most notable syntax requirements. Most of the other rules are imposed by the program itself (such as maximum delivery payment lag time of 60 days) and so will be discussed with the appropriate definitions of error messages.

Syntax for the data file:

Initial Line:

820101,0,6,25,.1

^	^	^	^	^	
					Profit rate as decimal
					Number of transactions in file
					Number of cost elements (max of 10)
					Start up cost (max 6 digits)
					Date contract begins

Title line (must be second)

CASH RUN PROJECT J-327

^	
	Run title (max 50 characters)

Cost element lines

[illegible]

Transaction lines

A	A	A	
			Progress payment rate (must be decimal number)
			Actual payment received(max of 6 digits)
			Date payment received.

Individual cost, one entry per cost element (max of 6 digits each)

#### 4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

---

##### Transaction Lines Cont'd

##### Type 4 (Planned delivery)

4,830201,1000

↑            ↑  
|            |  
|            |    Value of delivery (max of 6 digits)  
|            |    Planned date of delivery

##### Type 5 (Actual delivery)

5,820425,250,.9

↑            ↑            ↑  
|            |            |  
|            |            |    Liquidation rate (must be a decimal number)  
|            |            |    Value of actual delivery (max 6 digits)  
|            |            |    Date of actual delivery

Most terms were defined as part of section 2, creating the data file. The definitions here supplement the ones given earlier. These definitions are in alphabetical order.

ACCOUNTING PERIOD - The period over which costs are accumulated to be billed in a cost statement. The program is designed to compute progress payments no more frequently than monthly; however, a monthly cost accounting period may be a calendar month, 30 days, 4 weeks, or 5 weeks. Cost accounting periods must be at least 28 days in length and there may be no more than twelve per year. Used in type 3 transactions in the input data file.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D  
-----

ALTERNATE LIQUIDATION RATE - That portion of an actual delivery value which is assumed to have already been reimbursed. The alternate rate may only be used after 365 days of the contract has elapsed. See Ordinary Liquidation Rate, see DAR Appendix E, Part 5.

CASHII PROGRAM - The CASHII computer program calculates the highest whole number progress payment rate that is possible while keeping the contractor weighted cumulative work-in-process inventory from going below a level of 5.0%. The program is written in ANSI FORTRAN x3.9-1978.

CONTRACT START AND END DATES - The contract start date is the date, in year, month, day (YYMMDD) format, of the first day of the accounting period of the first cost statement. If actuals are input, this will be the day of incurred cost for the proposal. If there is no start-up cost, the contract start date will be the first day of the month of definitization. The contract end date is the date of the final delivery. Used in the input data file.

CONTRACTOR INVESTMENT RATIO (CIR) - Is calculated with weighted day dollars. Cumulative day dollars of costs reimbursed (CR) are subtracted from cumulative paid costs to get unreimbursed cost. CIR is then unreimbursed costs divided by cumulative work in process inventory. Work-in-process (WIP) is calculated as weighted cumulative paid cost (CP) less the weighted cumulative cost of deliveries made (COD). COD is weighted on the day of delivery payment and then removed from the weighted CP to arrive at a weighted WIP amount. Thus:

$$\begin{aligned} \text{WIP} &= \text{CP} - \text{COD} \\ \text{CIR} &= (\text{CP} - \text{CR}) / \text{WIP} \end{aligned}$$

CASHII seeks to hold CIR to not less than 5%. CASHII performs successive calculations to determine the highest progress rate ( up to 100% ) possible while keeping the weighted Contractor Investment Ratio equal to or greater than 5%. The result of the calculation is printed in the Analytical Report.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

-----

COST AS INCURRED - Cost as incurred is a type of cost which is eligible for progress payments when it is incurred. Incurred costs are those costs identified through the use of the accrual method of accounting and reporting. Facilities capital cost of money, which is recognized as an allowable cost, is also an incurred cost for progress payment purposes. Used in the input data file for appropriate cost elements. Use the code INCUR.

COST AS PAID - Cost as paid is a type of cost which is not eligible for progress payments until it is paid. Used in the input data file for appropriate cost elements. Use the code PAIDC.

COST ELEMENTS - Cost elements are those groupings of cost which are of the same progress payment eligibility and of similar float times. The model will accept up to 10 cost elements. Contractors are encouraged to develop computer programs to reformat DD Form 633 data to a format consistent with the DOD Cash Flow computer model input file requirements. All cost elements must be "typed" as one of INCUR, PAIDC or SUBCT. Must be specified in the input data file.

COST ELIGIBLE FOR CUSTOMARY PROGRESS PAYMENT - Includes only those recorded or estimated costs which result, at the time of the cost statement request, from payment made by cash, check, or other form of actual payment for items or services purchased directly for the contract, together with cost incurred, but not necessarily paid, for materials which have been issued from the Contractor's stores inventory and placed in the production process for use on the contract, for direct labor, for direct travel, for other direct in-house costs, and for properly allocable and allowable indirect costs. The CashII Flow Model converts incurred cost to paid cost by adding the cost element payment float to the incurred cost date (i.e., month-end date plus float).



#### 4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

---

**COST ELIGIBLE FOR 100% PROGRESS PAYMENT** - The amount of progress payments which have been paid to contractors' subcontractors and other divisions are eligible for 100% reimbursement. Used in the input data file for appropriate cost elements. Use the code SUBCT.

**COST STATEMENT OF MONTHLY COST** - The actual or projected monthly cost (by up to 10 cost elements) estimated to be incurred or paid during the cost accounting period. Specified in the input data file as transaction type 3. Must be entered in order by date.

**COST STATEMENT DATE** - The last day of the cost accounting period. Entered in the input data file as year, month, day, i.e., 820228.

**CUSTOMARY PROGRESS PAYMENT RATE** - The uniform standard percent (90% or 95%) or the flexible percent (if requested by the contractor) of the amount of the contractor's total costs. Calculated by CASHII or may be entered in response to the proper question when CASHII is executing.

**DATA FILE** - A collection of information stored in the computer. For the CASHII system the contract information discussed in section 2, necessary to perform the flexible progress payment calculations.

**DELIVERY PAYMENT LAG** - is the number of days between the delivery date and the receipt of the actual delivery payment. Specified in response to the question asked when CASHII is executed. May not exceed 60 days.

**ERROR MESSAGE** - A message supplied by CASHII indicating some problem exists which prevents normal program functions. For instance, entering a lag time which is beyond the maximum allowed by the program. See the listing of error messages which concludes this section.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

-----

FLEXIBLE PROGRESS PAYMENT RATE - The flexible progress payment rate is the progress payment rate calculated by the CASHII program. Once the contract is definitized this rate is maintained throughout the life of the contract. However, if actual and projected cash flow data generated during performance of the contract reveal that the progress payment rate will result in a weighted average investment in work-in-process inventory by the contractor in excess of 7% or less than 3% the progress payment rate shall be re-determined by using the DOD Cash Flow Computer model. In no event will the progress payment rate be less than the uniform standard progress payment rate or greater than 100%.

FLOAT - Float is the number of days between the cost statement date and the time the cost is paid on a cash basis. Bank lag times are included in the float number entered in the data file. The program will accept both positive and negative whole numbers for float entries. Specified for each cost element. See the examples of calculations in section 2.

MONTH OF DEFINITIZATION - The month of contract definitization is the month in which a contract is signed or definitized. This will be the first cost statement period.

ORDINARY LIQUIDATION RATE - That portion of an actual delivery value which is assumed to have been already reimbursed. Always equal to the progress payment rate. Used in the input data file, transaction type 5. See also Alternate Liquidation Rate, see DAR, Appendix E, Part 5.

PERCENT OF MATERIAL COST PAID - The percent of material cost that is paid as of the cost statement date. An estimate of the percentage of dollars paid on invoices per accounting period should be made and included in cost element description lines in the data file.

#### 4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

---

PROGRESS PAYMENT ELIGIBILITY CODES - There are three types of costs with different progress payment eligibility codes. These are: cost as incurred (INCUR), cost as paid (PAIDC) and cost eligible for 100% progress payment (SUBCT).

PROGRESS PAYMENT LAG - Progress payment lag is the number of days between the end of the cost statement date and the progress payment receipt. This lag should not normally include more than 5 working days from the cost statement date for the contractor to prepare and submit a progress payment request. Specified when executing CASHII. May not exceed 56 days.

START UP COST - Start-up cost is the cost authorized and incurred by the contractor prior to definitization, i.e., approval by the government of the signed contract. If progress payments have been received during this time start-up cost is calculated by totaling the cost to date of definitization and subtracting any progress payments received plus applicable profit. Start-up cost is input in the first line of the data file.

SUBCONTRACTOR PROGRESS PAYMENTS - Subcontractor progress payments are eligible for 100% reimbursement to the contractor. This is the only cost element that accepts negative whole numbers in a cost statement. These negative numbers represent subcontractor liquidations to accumulated progress payments when the subcontractor starts deliveries. The sum of subcontractor progress payments must equal 0 at the time of the last transaction.

#### 4. CASHII, TERMS, DEFINITIONS AND SYNTAX CONT'D

---

TRANSACTION TYPE - Each type of transaction used in the third section of the input data file must be identified uniquely. There are five types:

- 1) Actual progress payment received. These will be entered into the input data file as appropriate to reflect receipt of payment. Contains three items, date, amount, progress payment rate.
- 2) Planned progress payment. These transactions show up in the transaction report. They are calculated by and used only by the CASHII program.
- 3) Monthly cost statement (actual or estimated). Usually make up the bulk of the transactions entered in the input datafile. Contains date plus one entry for each cost element, zero if no value.
- 4) Planned delivery contains date and value.
- 5) Actual delivery. Contains date, value, liquidation rate.

All transactions must be entered in order by date. The total number of transaction lines must be equal to the number of transactions specified in the first line of the data file. The total number of transactions (including type 2's) may not exceed 325. If yours do, try combining earlier historical ones.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

---

UNLIQUIDATED PROGRESS PAYMENT - Column 10 on the Transaction report. It is that portion of cumulative progress payments which have not been liquidated.

WORK-IN-PROGRESS - A calculation performed by the program in order to calculate contractor investment ratio. WIP is calculated as cumulative day dollar cost of deliveries made subtracted from cumulative day dollar paid cost.

#### 4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

-----

##### ERROR MESSAGES

-----

As you might expect with computers, any number of things can keep it from working exactly right. CASHII tries hard to give you as much information as possible about the error. If the error is on a specific line, that line will be printed at the terminal. For instance, the following line has an error. Can you see it?

3,820330,150,50,100100,100,100

Right, the problem is the missing comma between the first two 100s. In this case the error message is:

Error in the following input line - then the line prints then a second message prints: The program was stopped because of an input file error.

and the program stops. Now you have to fix the problem and execute CASHII again. Sometimes when you get the error message above, there is nothing obviously wrong. The first thing to try is typing the entire line over again and saving the new version. Most times when an error occurs, CASHII continues checking the input data file for other errors.

#### 4. CASH TERMS, DEFINITIONS AND SYNTAX CONT'D

---

##### ERROR MESSAGES Cont'd

---

The following is the list of error messages with some comments on corrective action where it isn't obvious.

Actual progress payment must have a value greater than 0. The progress payment dated 'date' is less than 0.

'date' will be the date of the transaction.  
Fix the line and try again.

Cost and delivery transactions must not have value larger than six digits. The cost statement dated 'date' has a value that is too large.

'date' is the date of the type 3, 4 or 5 transaction in error. Could be a missing comma. If not divide all your input numbers by 10,100,1000,etc. to get smaller numbers.

Cost reimbursement cannot be greater than incurred cost. The transaction dated 'date' violates this rule. The delivery payments to date exceed the cost to date.

Total cost at this point from cost statements XXXXX

Total deliveries at this point XXXXX

Cost reimbursed at this point based on delivery prices XXXXX

'date' identifies the transaction where this error was noticed. Possibly a missing type 3 transaction. Could also be a type 4 or 5 transaction with too large a value.

4. CASHII, TERMS, DEFINITIONS AND SYNTAX CONT'D

-----

ERROR MESSAGES Cont'd

-----

Cost element 'X' does not have a valid cost element code.

'X' will be a number. The program starts counting at the first cost element. The number 6 would mean the sixth one down doesn't specify INCUR, SUBCT or PAIDC.

Data is not in chronological order. Check transactions dated 'date' and 'date' to correct.

'date' indicates transactions where the problem was noticed. Possibly a typo or you just don't have all the transactions in order by date.

Data file contains more actual progress payments than cost statements. The number of actual progress payments should equal the number of cost statements before the last actual progress payment.

could it be you accidentally wiped out a cost statement? Count them. Or maybe you didn't mean to enter an actual progress payment.

Data file indicated 'X' transactions were included in the data file. 'Y' were actually included.

'X' and 'Y' will be different numbers. They need to be the same. Your file said (in the first line) that 13 transactions were included, the program only found 12. Possibly a typo or maybe you forgot one. Fix and re-run

NOTE: If 'X' is smaller than 'Y', the program simply ignores the excess data.



#### 4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

---

##### ERROR MESSAGES Cont'd

---

Delivery payment must have a value larger than 0 but not contain more than 6 digits. The transaction dated 'date' violates this rule.

fix it. If the value is too big divide by 10,100, etc. Remember, the rest of the numbers in the file must be consistent.

Do not input any planned progress payments into the data file. The transaction dated 'date' must be removed.

'date' indicates the transaction in error. You have a type 2 transaction in the data file. You can't do that, only the program uses type 2 transactions.

First transaction must be a cost report.

you have to have a type 3 transaction as the first transaction in your data file.

Float days on costs paid in the present accounting period should be negative. This is not the case for cost element 'X'.

'X' identifies the cost element in question.

Liquidation rate associated with an actual delivery must be between 0 and 1. The transaction dated 'date' violates this rule.

'date' indicates the transaction in error. Liquidation rate must be a decimal number. A 90% rate is specified as .9.

#### 4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

---

##### ERROR MESSAGES Cont'd

---

Last transaction must be a delivery.

yours isn't, it should be. If it's there check the total number of transactions in the file and make sure you have the proper number in the first line of the file.

Negative cost is permitted only for subcontractor type progress payments. The cost statements dated 'date' has a negative value and is not a subcontractor cost element.

probably a typo.

Number of cost elements must be between 1 and 10.

combine similar costs into one category.

Percent paid in the present accounting period should be between 0 and 1 for cost element 'X'.

'X' identifies the cost element in question, counting down from the first cost element. This value must be a decimal number. 80% would be .8.

Planned progress payments cannot occur less than 28 days apart. The transaction dated 'date' violates this rule.

check input file to insure cost statements are at least 28 days apart.

Progress payment rate must be between 0 and 1. The transaction dated 'date' violates this rule.

date identifies the transaction in error. You must specify the rate as a decimal. 90% would be .9.

#### 4. CASHII, TERMS, DEFINITIONS AND SYNTAX CONT'D

##### ----- ERROR MESSAGES CONT'D -----

The profit rate computed from the data file (XX.XX) does not match the profit rate (XX.XX) in the first line of the data file.

Computed profit based on the following values computed from data inputs:

Total price from delivery prices	XXXXXX
Total cost from cost statement	XXXXXX
Profit dollars	XXXXX

Profit rate must be between 0 and 1.

a mistake on the first line of your data file. The last number is the profit rate and it has to be a decimal number. 10% profit is specified as .1.

Start-up cost cannot be a negative number.

fix it.

Subcontractor progress payments should add to 0 across the life of the contract for cost element 'X'.

subcontractor costs must be fully liquidated.  
'X' identifies which cost element has the problem.  
Fully liquidated means add to zero. You don't have enough negative numbers to equal the total value of subcontractor expenses.

Transaction dated 'date' has an illegal transaction code. The code must be a 1, 2, 3, 4, or 5.

actually, it can't be a 2 either.

The transaction date 'date' has an illegal month 'XX'.

The month is less than zero or greater than 14.

The transaction date 'date' has an illegal day 'XX'.

The day is greater than the number of days in a particular month.

The float days for unpaid costs cannot exceed 28 days. This is not the case for cost element 'X'.

See note at bottom of page 19.

# SAMPLE USING GEISCO

HH  
 UF=MDG36P21  
 PASSWORD  
 PASSWORD

IDENTIFY TERMINAL TYPE AND SPEED  
 ENTER YOUR USER NUMBER  
 ENTER YOUR PASSWORD

READY  
 NEW SAMP1  
 SYSTEM- F77

NEW IS THE COMMAND NECESSARY TO ENTER  
 NEW DATA. SAMP1 IS THE FILE NAME.  
 F77 MEANS FORTRAN

READY  
 100 8201P1,P,6,22,1  
 110 CASH EXAMPLE WITH PROJECTED DATA  
 120 DIRECT LABOR,INCR,-2,0,0  
 130 OVERHEAD,INCR,15,0,0  
 140 SUBCONT PROG PAY,20,8,-2  
 150 MATERIAL,PAIDC,20,8,-2  
 160 PURCHASED PARTS,PAIDC,20,9,-1  
 170 INTERDIVISIONAL,PAIDC,15,9,-4  
 180 3,82013P,P,0,100,0,0  
 190 3,82022S,0,0,100,0,0  
 200 3,82030F,150,50,0,0,0  
 210 1,8204DELETED  
 210 4,820405,050  
 220 3,820430,150,50,100,100,100  
 230 3,820502,150,50,100,100,100  
 END DES

ACCIDENTALLY TYPED 2 R'S  
 ← IS THE CORRECTION KEY.

1,8204 IS A MISTAKE. HOLD DOWN CONTROL  
 KEY AND TYPE AN X. THE COMPUTER PRINTS  
 DELETED. START THE LINE OVER AGAIN.

FINISH ENTERING YOUR DATA

WHEN FINISHED ENTERING YOUR DATA,  
 REMOVE LINE NUMBERS FROM THE FILE

READY  
 REP

SAVE (REPLACE) THE DATA

READY  
 RUN CASH11

EXECUTE THE PROGRAM

CASH11 TIME DATE

PAPER MOVES TO TOP OF PAGE (IF YOUR  
 TERMINAL IS EQUIPPED PROPERLY)

\*\*\*\*\*  
 \*\*\* DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL \*\*\*  
 \*\*\* VERSION DATED FEBRUARY 1, 1983 \*\*\*  
 \*\*\*\*\*

NOTE: COMMAND SYSTEM REFERENCE MANUAL  
 PUBLICATION #3501.01 FURTHER  
 EXPLAINS COMMANDS.

CAUTION--THE CASH MODEL VERSION IS NOT COMPATIBLE WITH  
 THE USERGUIDE DATED AUGUST 28,1981. PLEASE CONSULT THE  
 USERGUIDE DATED FEBRUARY 1,1983 FOR GUIDANCE

INPUT FILE NAME?SAMP1

ENTER THE NAME OF YOUR DATA FILE

to the bcs network  
access port is bcs 0ed

select desired service: CTS

mainstream-cts online

SAMPLE USING BCS

CPXLOGON AFSPAP1

ENTER PASSWORD:

CPX

ENTER YOUR USER NUMBER

ENTER YOUR PASSWORD

LOGMSG - 09:40:29 EST WEDNESDAY 01/26/83

\*\* PLEASE TYPE NEWS MEMO 428 - RAMIS II - NEW DOCUMENTS AND INCREASED \*

\*\* MACHINE SIZE REQUIREMENTS \*\*

\*\* PLEASE TYPE NEWS MEMO 417 - INTELLECT AVAILABLE \*\*

\*\* PLEASE TYPE NEWS MEMO 435 - PLOTTING SERVICES AVAILABLE \*\*

\*\* PLEASE TYPE NEWS MEMO 436 - PHONE NUMBER ADDED - SIOUX FALLS, SD \*\*

\*\* PLEASE TYPE NEWS MEMO 421 - TELL-A-CRAF VERSION 4.1 AVAILABLE \*\*

\*\* PLEASE TYPE NEWS MEMO 437 - PHONE NUMBER OUTAGE - PHOENIX, AZ. 1/28 \*\*

\*\* PLEASE TYPE NEWS MEMO 434 - IMPROVED LOCAL DIAL FOR RJE SERVICES \*\*

LOGON AT 07:00:09 EST FRIDAY 01/28/83 LINE 606 (2-1-0ED)

CMS/SP REL 1 01/25/83 V009

D (292) R/O

ENTER PROJECT ID:

>000

Z (199) R/O

R;

ENTER YOUR PROJECT ID

R;

EDIT IS A COMMAND, NECESSARY TO

ENTER DATA

SAMP1 IS THE FILE NAME

DATA IS THE FILE TYPE

I MEANS WE WANT TO INSERT DATA

CXEDIT SAMP1 DATA

EXI

I>000101.0,6,22,1

I>CASH EXAMPLE WITH PROJECTED DATA

I>DIRECT LABOR,INCL,2,0,0

I>OVERHEAD,INCL,15,0,0

I>SUBCONT PROG PAY,SUBCT,20,8,-2

I>MATERIAL,PAIDC,28,8,-2

I>PURCHASED PARTS,PAIDC,20,9,-1

I>INTERDIVISIONAL,PAIDC,15,9,-4

I>3,820130,0,0,0,100,0,0

I>3,820220,0,0,0,100,0,0

I>3,820330,150,50,0,0,0,0

I>1,820404,24,820425,250

I>3,820430,150,50,100,100,100,100

I>3,820530,150,50,100,100,100,100

I>3,820630,150,50,100,100,100,100

I>

EXFILE

R;

ENTERING THE DATA

HIT THE M INSTEAD OF , @ IS THE  
CORRECTION KEY 1,820404,2 WAS A MISTAKE,  
HOLD DOWN THE CONTROL KEY, TYPE AN X  
AND CONTINUE WITH THE CORRECT DATA.

ENTERING A CARRIAGE RETURN WHEN NO MORE  
DATA

SAVES THE FILE

EXECUTE THE PROGRAM

ENTER THE NAME OF YOUR DATA FILE

CXCASHI

ENTER INPUT FILENAME

>SAMP1

NOTE: MAINSTREAM - CTS COMMAND LANGUAGE  
REFERENCE MANUAL FURTHER EXPLAINS  
THE COMMANDS.

REFERENCE MATERIAL FOR THIS DOCUMENT IS  
CONTAINED IN DEFENSE ACQUISITION REGULATION,  
APPENDIX E, PART 5.

```

      INTEGER STDATE,NMCSTL,NMTRAN,ACFRG,LSTDEL,
      &PRGLAG,DELLAG,PRGFLG,ALTLDY,FLOATD,ALQFLG,SUBCNT,
      &UPDENT,STDAY,PFENT,CNTST,TWOFLG,CSTDAY(170),DATHEY
      REAL TRNSAC(325,20),PRFRAT,PROFIT,ELMENT(10,5),OUTPUT(325,10),
      &PRGRTE,RA110,TOTFYD,TODAYS,STCOST
      REAL CSTFLT(1000,2),UPDFLT(1000,2),UPDSUB(1000,2)
      REAL CCOST,TOPROF,TOTCOS,SUMCST,TOTCHG
      CHARACTER*20 COSTEL(10,2)
      CHARACTER*50 TITLE
      CHARACTER*3 TRNFLG,CSTFLG,FLGIT,FLXFLG,ERROR,DATCHK
      LOGICAL*1 CC
      DATA CC / ZOC /
C *****
C      SECTION ABOVE ESTABLISHES ATTRIBUTES OF VARIABLES PASSED
C      BETWEEN MAIN PROGRAM AND SUBROUTINES.
C      SECTION IMMEDIATELY BELOW WRITES MODEL HEADINGS.
C *****
10  FORMAT(1H1)
20  FORMAT(' ***** )
30  FORMAT(' *** DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL *** )
40  FORMAT(' ***          VERSION DATED FEBRUARY 1, 1983          ***')
50  FORMAT(' CAUTION--THIS CASH MODEL VERSION IS NOT COMPATIBLE WITH )
51  FORMAT(' THE USER GUIDE DATED AUGUST 28,1981. PLEASE CONSULT THE )
52  FORMAT(' USER GUIDE DATED FEBRUARY 1,1983 FOR GUIDANCE. ')
60  FORMAT( / )
70  FORMAT( / )
      WRITE(6,FMT='(A1)' ) CC
      WRITE(6,20)
      WRITE(6,30)
      WRITE(6,40)
      WRITE(6,20)
      WRITE(6,70)
      WRITE(6,50)
      WRITE(6,51)
      WRITE(6,52)
      WRITE(6,70)
      RERUN=0
100  ACFRG=0
C *****
C      CALLS SUBROUTINE TO READ INPUT FILE.
C *****
      CALL FILEIN(STDATE,STCOST,NMCSTL,NMTRAN,PRFRAT,TITLE,COSTEL,
      &ELMENT,TRNSAC,ACFRG,STDAY,TRNCNT,ERROR)
      IF (ERROR.EQ. 'YES') THEN
          WRITE(6,200)
          GO TO 2000
      END IF
C *****
C      CALLS SUBROUTINE TO EDIT INPUT DATA.
C *****
      DATCHK='NO'
      DATHEY=0
      CALL EDIT(TRNSAC,ELMENT,NMCSTL,STDATE,STCOST,PRFRAT,NMTRAN
      &,COSTEL,ERROR,CCOST,ACFRG,TOTFYD,TOPROF,PROFIT,LSTDEL,TODAYS,
      &TOTCOS)
      IF (ERROR.EQ. 'YES') THEN
200  FORMAT(' THE PROGRAM WAS STOPPED BECAUSE OF AN INPUT FILE ERROR.
      300

```

```

CAS00010
CAS00020
CAS00030
CAS00040
CAS00050
CAS00060
CAS00070
CAS00080
CAS00090
CAS00100
CAS00110
CAS00120
CAS00130
CAS00140
CAS00150
CAS00160
CAS00170
CAS00180
CAS00190
CAS00200
CAS00210
CAS00220
CAS00230
CAS00240
CAS00250
CAS00260
CAS00270
CAS00280
CAS00290
CAS00300
CAS00310
CAS00320
CAS00330
CAS00340
CAS00350
CAS00360
CAS00370
CAS00380
CAS00390
CAS00400
CAS00410
CAS00420
CAS00430
CAS00440
CAS00450
CAS00460
CAS00470
CAS00480
CAS00490
CAS00500
CAS00510
CAS00520
CAS00530
CAS00540
CAS00550
CAS00560
CAS00570
CAS00580

```

WRITE(6,200)	CAS00590
WRITE(6,70)	CAS00600
*****	CAS00610
ROUTINE ALLOWS USER TO CHECK INPUT DATA WHEN EDIT ROUTINE	CAS00620
DETECTS ERRORS.	CAS00630
*****	CAS00640
FORMAT(' WOULD YOU LIKE TO CHECK YOUR INPUT DATA? (YES OR NO)')	CAS00650
WRITE(6,210)	CAS00660
READ(5,FMT='(A3)') DATCHK	CAS00670
IF(DATCHK(1:1).NE.'Y'.AND.DATCHK(1:1).NE.'N') THEN	CAS00680
GO TO 205	CAS00690
END IF	CAS00700
IF(DATCHK.EQ.'YES') THEN	CAS00710
FORMAT(' DO YOU WANT TO CHECK: 1-COST INPUT DATA; 2-DELIVERY INPUT	CAS00720
& DATA; OR 3-BOTH?')	CAS00730
WRITE(6,70)	CAS00740
WRITE(6,220)	CAS00750
READ(5,FMT=*) DATKEY	CAS00760
IF(DATKEY.LT.1.OR.DATKEY.GT.3) THEN	CAS00770
WRITE(6,70)	CAS00780
GO TO 225	CAS00790
END IF	CAS00800
END IF	CAS00810
IF(DATKEY.GT.0) THEN	CAS00820
GO TO 305	CAS00830
ELSE	CAS00840
GO TO 2000	CAS00850
END IF	CAS00860
GO TO 2000	CAS00870
END IF	CAS00880
*****	CAS00890
CALLS SUBROUTINE TO GET USER INPUTS FROM TERMINAL.	CAS00900
*****	CAS00910
CALL DATAIN(PRGLAG,DELLAG,FLXFLG,PRGRTE,PRGFLG,	CAS00920
SALTLDY,TRNFLG,CSTFLG,RERUN,TWOFLG)	CAS00930
*****	CAS00940
CALLS SUBROUTINE TO CREATE PLANNED PROGRESS PAYMENTS AND	CAS00950
ADD DELIVERY LAG TO DELIVERY DATES.	CAS00960
*****	CAS00970
CALL CREPAY(TRNSAC,NMTRAN,ACPRG,CCOST,PRGLAG,PRGFLG,STDAY,DELLAG,	CAS00980
ERROR)	CAS00990
IF (ERROR.EQ.'YES') THEN	CAS01000
GO TO 205	CAS01010
END IF	CAS01020
*****	CAS01030
INITIALIZES (OR RESETS) VARIABLES TO ZERO.	CAS01040
INITIALIZES FIRST TIME THROUGH; RESETS AS FLEX RATE STEPS	CAS01050
DOWN AND WHEN ALTERNATIVES ARE RUN WITH THE SAME DATA	CAS01060
FILE.	CAS01070
*****	CAS01080
230 DO 250 I=1,NMTRAN	CAS01090
DO 240 J=1,10	CAS01100
OUTPUT(I,J)=0	CAS01110
CONTINUE	CAS01120
CONTINUE	CAS01130
DO 270 I=1,1000	CAS01140
DO 260 J=1,2	CAS01150
CSTFLT(I,J)=0	CAS01160
UPDFLT(I,J)=0	CAS01170
UPDSUB(I,J)=0	CAS01180



7	CONTINUE	CAS01190
8	CONTINUE	CAS01200
	DO 290 I =1,300	CAS01210
	DO 280 J=14,17	CAS01220
	TRANSAC(I,J)=0	CAS01230
80	CONTINUE	CAS01240
85	CONTINUE	CAS01250
	DO 295 I=1,170	CAS01260
	CSTDAY(I)=0	CAS01270
895	CONTINUE	CAS01280
	LFLG=0	CAS01290
	PPCNT=0	CAS01300
	CNTST=0	CAS01310
	ALTFRF=0	CAS01320
	COSDEL=0	CAS01330
	FDCOST=0	CAS01340
	INFLOW=0	CAS01350
	COUNT3=0	CAS01360
	FLOATD=0	CAS01370
	ALOFLG=0	CAS01380
	SUBCNT=0	CAS01390
	UPDCNT=0	CAS01400
	SUMCST=0	CAS01410
	TOTCHG=0	CAS01420
	*****	CAS01430
	CALLS SUBROUTINES TO HANDLE EACH TRANSACTION TYPE.	CAS01440
	*****	CAS01450
	DO 300 I=1,NMTRAN	CAS01460
	IF (TRANSAC(I,1).EQ.1) THEN	CAS01470
	CALL ACTFPY(TRANSAC,I,OUTPUT,FLOATD,CSTFLT,PRFRAT,ALTLDY,STCOST	CAS01480
	,&,PPCNT)	CAS01490
	END IF	CAS01500
	IF (TRANSAC(I,1).EQ.2) THEN	CAS01510
	CALL PRGPAY(TRANSAC,I,PRGRTE,CSTFLT,FLOATD,OUTPUT,TOTCHG,	CAS01520
	&SUMCST,TOTPYD,ALTLDY,PRFRAT,STCOST,PPCNT,CSTDAY,TWOFLG)	CAS01530
	END IF	CAS01540
	IF (TRANSAC(I,1).EQ.3) THEN	CAS01550
	CALL COSTST(TRANSAC,I,NMCSTL,ELMENT,FLOATD,CSTFLT,OUTPUT,	CAS01560
	&SUBCNT,UPDCNT,UPDFLT,UPDSUB,PRGRTE,STCOST,CNTST,CSTDAY,	CAS01570
	&TWOFLG)	CAS01580
	END IF	CAS01590
	IF (TRANSAC(I,1).EQ.4) THEN	CAS01600
	CALL PLNDEL(TRANSAC,I,ALTLDY,PRFRAT,ALOFLG,CSTFLT,FLOATD,	CAS01610
	&TOTCHG,SUMCST,OUTPUT,PRGRTE,LSTDEL,DELLAG,STCOST,TODAYS,SDAY,	CAS01620
	&ALTFRF,LFLG)	CAS01630
	END IF	CAS01640
	IF (TRANSAC(I,1).EQ.5) THEN	CAS01650
	CALL ACTDEL(TRANSAC,I,PRFRAT,ALTLDY,ALOFLG,CSTFLT,FLOATD,	CAS01660
	&TOTCHG,SUMCST,OUTPUT,LSTDEL,DELLAG,STCOST,TODAYS,SDAY,ALTFRF)	CAS01670
	END IF	CAS01680
90	CONTINUE	CAS01690
	*****	CAS01700
	CALLS SUBROUTINE TO COMPUTE CONTRACTOR INVESTMENT RATIO.	CAS01710
	*****	CAS01720
	CALL INVEST(TRANSAC,NMTRAN,TODAYS,CSTFLT,FLOATD,RATIO,OUTPUT,DELLAG	CAS01730
	&,STCOST,PRFRAT)	CAS01740
	IF (FLXFLG(1:1).EQ.'N') THEN	CAS01750
	IF (RATIO.LT.(.05).AND.PRGRTE.GT.(.90)) THEN	CAS01760
	PRGRTE=PRGRTE-0.01	CAS01770
	GO TO 230	CAS01780

```

      END IF
      END IF
      *****
      CALLS SUBROUTINE TO PRINT REPORTS.
      *****
5    CALL PRINTOUT (TRANFLG, COSTFLG, PRGFLG, TRANSAC, ELEMENT, FLXFLG, NMTRAN,
      NMOSTL, STDATE, COSTEL, PRFRAT, STCOST, TOTPYD, TOTCOS, OUTPUT, TORPROF,
      PROFIT, PRGRTE, RATIO, PRGLAG, DELLAG, LSTDEL, TITLE, DATKEY, LFLG)
      *****
      ALLOWS USER TO CHANGE TERMINAL INPUTS AND RE-RUN USING
      THE SAME DATA FILE.
      *****
10   FORMAT(' WOULD YOU LIKE TO RUN THE MODEL AGAIN USING THE SAME ')
11   FORMAT(' DATA FILE BUT CHANGING THE VARIABLES AT THE TERMINAL' )
12   FORMAT(' INPUT? (YES / NO) ')
13   WRITE(6,FMT='(A1)') CC
      WRITE(6,310)
      WRITE(6,311)
      WRITE(6,312)
      READ(5,FMT='(A3)') FLGIT
      IF (FLGIT(1:1).NE.'Y').AND.(FLGIT(1:1).NE.'N') THEN
        GO TO 320
      END IF
      IF (FLGIT.EQ.'YES') THEN
        REFRUN=1
        GO TO 100
      END IF
14   CONTINUE
      STOP
      END
      *****          END OF MAIN PROGRAM          *****

      *****          START OF SUBROUTINE FILEIN          *****
      SUBROUTINE FILEIN(STDATE,STCOST,NMOSTL,NMTRAN,PRFRAT,TITLE,COSTEL,
      &ELEMENT,TRANSAC,ACPRG,STDAY,TRNCNT,ERROR)
      REAL PRFRAT,ELEMENT(10,5),TRANSAC(325,20),STCOST
      INTEGER A,TRNCNT,STDATE,NMOSTL,NMTRAN,ACPRG,STDAY,IPDS,IERR
      CHARACTER*20 COSTEL(10,2)
      CHARACTER*50 TITLE
      CHARACTER*80 HOLD,SAVE
      CHARACTER*8 SHOLD
      CHARACTER*3 ERROR
      DIMENSION KK1(5)
      DATA KK1/4,3,8,3,4/
      ERROR='NO'
      OPEN(UNIT=1,STATUS='OLD',ACCESS='SEQUENTIAL')
      *****
      READS FIRST LINE OF INPUT FILE.
      *****
      READ(UNIT=1,FMT=*) STDATE,STCOST,NMOSTL,NMTRAN,PRFRAT
      READ(UNIT=1,FMT=*) TITLE
      READ(UNIT=1,FMT='(A80)') HOLD
      SAVE=HOLD
      CALL GETSTA(HOLD, , ,SHOLD,IPDS)
      IF (IPDS.LT.2.OR.IPDS.GT.9) GOTO 135
      STDATE=INTSTV(SHOLD,IERR)
      IF (IERR.EQ.1) GOTO 135

```

CAS01790  
 CAS01800  
 CAS01810  
 CAS01820  
 CAS01830  
 CAS01840  
 CAS01850  
 CAS01860  
 CAS01870  
 CAS01880  
 CAS01890  
 CAS01900  
 CAS01910  
 CAS01920  
 CAS01930  
 CAS01940  
 CAS01950  
 CAS01960  
 CAS01970  
 CAS01980  
 CAS01990  
 CAS02000  
 CAS02010  
 CAS02020  
 CAS02030  
 CAS02040  
 CAS02050  
 CAS02060  
 CAS02070  
 CAS02080  
 CAS02090  
 CAS02100  
 CAS02110  
 CAS02120  
 CAS02130  
 CAS02140  
 CAS02150  
 CAS02160  
 CAS02170  
 CAS02180  
 CAS02190  
 CAS02200  
 CAS02210  
 CAS02220  
 CAS02230  
 CAS02240  
 CAS02250  
 CAS02260  
 CAS02270  
 CAS02280  
 CAS02290  
 CAS02300  
 CAS02310  
 CAS02320  
 CAS02330  
 CAS02340  
 CAS02350  
 CAS02360  
 CAS02370  
 CAS02380

CALL GETSTR(HOLD, ' ', SHOLD, IPOS)	CAS02390
IF (IPOS.LT.2.OR.IPOS.GT.9) GOTO 135	CAS02400
51COST=REALSV(SHOLD, IERR)	CAS02410
IF (IERR.EQ.1) GOTO 135	CAS02420
CALL GETSTR(HOLD, ' ', SHOLD, IPOS)	CAS02430
IF (IPOS.LT.2.OR.IPOS.GT.9) GOTO 135	CAS02440
NMCSTL=INTSTV(SHOLD, IERR)	CAS02450
IF (IERR.EQ.1) GOTO 135	CAS02460
CALL GETSTR(HOLD, ' ', SHOLD, IPOS)	CAS02470
IF (IPOS.LT.2.OR.IPOS.GT.9) GOTO 135	CAS02480
NMTRAN=INTSTV(SHOLD, IERR)	CAS02490
IF (IERR.EQ.1) GOTO 135	CAS02500
CALL GETSTR(HOLD, ' ', SHOLD, IPOS)	CAS02510
IF (IPOS.GT.9) GOTO 135	CAS02520
FFFRAT=REALSV(SHOLD, IERR)	CAS02530
IF (IERR.EQ.1) GOTO 135	CAS02540
GOTO 140	CAS02550
35 WRITE(6,390)SAVE	CAS02560
ERROR= YES	CAS02570
GOTO 600	CAS02580
40 CONTINUE	CAS02590
*****	CAS02600
READS TITLE FROM INPUT FILE.	CAS02610
*****	CAS02620
READ(UNIT=1,FMT='(A80)')HOLD	CAS02630
TITLE=HOLD	CAS02640
DO 120 I=1,NMCSTL	CAS02650
DO 110 J = 1,5	CAS02660
ELMENT(I,J)=0	CAS02670
110 CONTINUE	CAS02680
120 CONTINUE	CAS02690
*****	CAS02700
READS COST ELEMENT DATA FROM INPUT FILE.	CAS02710
*****	CAS02720
DO 190 I=1,NMCSTL	CAS02730
READ(UNIT=1,FMT=*) COSTEL(I,1),COSTEL(I,2),ELMENT(I,2),ELMENT(I,	CAS02740
3),ELMENT(I,4)	CAS02750
READ(UNIT=1,FMT='(A80)')HOLD	CAS02760
SAVE=HOLD	CAS02770
CALL GETSTR(HOLD, ' ', COSTEL(I,1), IPOS)	CAS02780
IF (IPOS.LT.2) GOTO 185	CAS02790
CALL GETSTR(HOLD, ' ', SHOLD, IPOS)	CAS02800
IF (IPOS.LT.2.OR.IPOS.GT.9) GOTO 185	CAS02810
COSTEL(I,2)=SHOLD	CAS02820
CALL GETSTR(HOLD, ' ', SHOLD, IPOS)	CAS02830
IF (IPOS.LT.2.OR.IPOS.GT.9) GOTO 185	CAS02840
ELMENT(I,2)=REALSV(SHOLD, IERR)	CAS02850
IF (IERR.EQ.1) GOTO 185	CAS02860
CALL GETSTR(HOLD, ' ', SHOLD, IPOS)	CAS02870
IF (IPOS.LT.2.OR.IPOS.GT.9) GOTO 185	CAS02880
ELMENT(I,3)=REALSV(SHOLD, IERR)	CAS02890
IF (IERR.EQ.1) GOTO 185	CAS02900
CALL GETSTR(HOLD, ' ', SHOLD, IPOS)	CAS02910
IF (IPOS.GT.9) GOTO 185	CAS02920
ELMENT(I,4)=REALSV(SHOLD, IERR)	CAS02930
IF (IERR.EQ.1) GOTO 185	CAS02940
IF (COSTEL(I,2).EQ.'INCUR') THEN	CAS02950
ELMENT(I,1)=1	CAS02960
END IF	CAS02970
IF (COSTEL(I,2).EQ.'PAIDC') THEN	CAS02980

ELMENT(I,1)=2	CAS02990
END IF	CAS03000
IF (COSTEL(I,2).EQ.'SUBCT') THEN	CAS03010
ELMENT(I,1)=3	CAS03020
END IF	CAS03030
GO TO 190	CAS03040
185 WRITE(6,390) SAVE	CAS03050
ERROR='YES'	CAS03060
190 CONTINUE	CAS03070
***** NEXT LINE ESTABLISHES RELATIVE REFERENCE POINT FOR CONTRACT	CAS03080
START DATE. ALL TRANSACTIONS WILL USE THIS POINT	CAS03090
TO DETERMINE DAYS AFTER CONTRACT AWARD.	CAS03100
STDAY= KDAY(STDATE)	CAS03110
KK1(3)= NMSTL + 2	CAS03120
TRNCNT=NMTRAN*2	CAS03130
DO 270 I=1,TRNCNT	CAS03140
DO 260 J=1,20	CAS03150
TRANSAC(I,J)=0	CAS03160
260 CONTINUE	CAS03170
270 CONTINUE	CAS03180
*****	CAS03190
READS TRANSACTION DATA FROM INPUT FILE.	CAS03200
*****	CAS03210
DO 330 I=1,NMTRAN+1	CAS03220
READ(UNIT=1,FMT= (A80)',END=500)HOLD	CAS03230
SAVE=HOLD	CAS03240
CALL GETSTR(HOLD,' ',SHOLD,IP0S)	CAS03250
IF (IP0S.NE.2) GOTO 290	CAS03260
A=INTSTV(SHOLD,IERR)	CAS03270
IF (IERR.EQ.1) GOTO 290	CAS03280
IF (A.LT.1.OR.A.GT.5) GOTO 290	CAS03290
TRANSAC(I,1) = A	CAS03300
DO 280 J=2,FF1(A)	CAS03310
CALL GETSTR(HOLD,' ',SHOLD,IP0S)	CAS03320
IF (IP0S.EQ.0.AND.(J.NE.KK1(A))) GOTO 290	CAS03330
IF (IP0S.GT.7.OR.(IP0S.LT.2.AND.J.NE.KK1(A))) GOTO 290	CAS03340
TRANSAC(I,J)=REALSV(SHOLD,IERR)	CAS03350
IF (IERR.EQ.1) GOTO 290	CAS03360
IF (J.EQ.2) THEN	CAS03370
F=INT(TRANSAC(I,2))	CAS03380
TRANSAC(I,13) = (KDAY(F)-STDAY)	CAS03390
ENDIF	CAS03400
GOTO 280	CAS03410
190 WRITE(6,390) SAVE	CAS03420
ERROR='YES'	CAS03430
GOTO 330	CAS03440
390 FORMAT(' ERROR IN FOLLOWING INPUT LINE',/,A80)	CAS03450
280 CONTINUE	CAS03460
330 CONTINUE	CAS03470
*****	CAS03480
CHECKS TO SEE IF PROGRAM READ THE NUMBER OF TRANSACTIONS	CAS03490
STATED IN THE FIRST LINE OF THE INPUT FILE.	CAS03500
*****	CAS03510
IF (1.LE.NMTRAN) THEN	CAS03520
WRITE(6,400) NMTRAN,I-1	CAS03530
ERROR='YES'	CAS03540
END IF	CAS03550
400 FORMAT(' THE DATA FILE INDICATED ',I2,' TRANSACTIONS WERE INCLUDED	CAS03560
IN THE FILE. ONLY ',I2,' TRANSACTIONS ARE ACTUALLY INCLUDED. PLEASE CHECK INPUT FILE.')	CAS03570
	CAS03580

```

CDEE (UNIT=1)
RETURN
END

```

```

*****      END OF SUBROUTINE FILEIN      *****

```

```

*****      START OF SUBROUTINE EDIT      *****

```

```

*****      EACH EDIT CHECK IS UNDERLINED WITH *'S.      *****

```

```

SUBROUTINE EDIT(TRANSAC,ELMENT,NMCSTL,STDATE,STCOST,PRFRAT,NMTRAN,
COSTEL,ERROR,CCOST,ACPRG,TOTPYD,TOPROF,PROFIT,LSTDEL,TODAYS,
%TUTLOS)

```

```

REAL TRANSAC(325,20),ELMENT(10,5),PRFRAT,CCOST,TOPROF,PROFI
%T,TOTLOS,TOTPYD,STCOST,TODAYS

```

```

INTEGER NMCSTL,STDATE,NMTRAN,LSTDEL,DAFMO(12)

```

```

CHARACTER*20 COSTEL(10,2)

```

```

CHARACTER*3 ERROR

```

```

INTEGER CACPRG,ACPRG

```

```

REAL COSTPD,P,PL,TH

```

```

DATA DAFMO/31,29,31,30,31,30,31,31,30,31,30,31/

```

```

ERROR= NO

```

```

TOTPYD=0

```

```

CCOST=0

```

```

CACPRG=0

```

```

TOTCOS=STCOST

```

```

IF (STCOST.LT.0) THEN

```

```

10  FORMAT( ' THE START-UP COST CANNOT BE A NEGATIVE NUMBER.  / ')
WRITE(6,10)

```

```

ERROR= YES

```

```

WRITE(6,4000)

```

```

END IF

```

```

IF (NMCSTL.GT.10.OR.NMCSTL.LT.1) THEN

```

```

10  FORMAT( ' THE NUMBER OF COST ELEMENTS MUST BE BETWEEN 1 AND 10
/ ')

```

```

WRITE(6,20)

```

```

ERROR= YES

```

```

WRITE(6,4000)

```

```

END IF

```

```

IF (PRFRAT.LT.0.OR.PRFRAT.GT.1) THEN

```

```

10  FORMAT( ' THE PROFIT RATE MUST BE BETWEEN 0 AND 1.  / ')
WRITE(6,30)

```

```

ERROR= YES

```

```

WRITE(6,4000)

```

```

END IF

```

```

DO 800 I=1,NMTRAN

```

```

IYRPC=INT(TRANSAC(I,2))/10000

```

```

ISFLTY=IYRPC * 10000

```

```

ISFLTM=INT(TRANSAC(I,2)) - ISFLTY

```

```

IMOPC=ISFLTM/100

```

```

ISFLTD=IMOPC * 100

```

```

IMOPC=ISFLTM - ISFLTD

```

```

IF (IMOPC.LT.1.OR.IMOPC.GT.12) THEN

```

```

10  FORMAT( ' THE TRANSACTION DATE .E.G., HAS AN ILLEGAL MONTH',
/ ')

```

```

WRITE(6,32) TRANSAC(I,2), IMOPC

```

```

ERROR= YES

```

```

WRITE(6,4000)

```

```

END IF

```

```

END DO

```

```

CAS03540
CAS03550
CAS03560
CAS03570
CAS03580
CAS03590
CAS03600
CAS03610
CAS03620
CAS03630
CAS03640
CAS03650
CAS03660
CAS03670
CAS03680
CAS03690
CAS03700
CAS03710
CAS03720
CAS03730
CAS03740
CAS03750
CAS03760
CAS03770
CAS03780
CAS03790
CAS03800
CAS03810
CAS03820
CAS03830
CAS03840
CAS03850
CAS03860
CAS03870
CAS03880
CAS03890
CAS03900
CAS03910
CAS03920
CAS03930
CAS03940
CAS03950
CAS03960
CAS03970
CAS03980
CAS03990
CAS04000
CAS04010
CAS04020
CAS04030
CAS04040
CAS04050
CAS04060
CAS04070
CAS04080
CAS04090
CAS04100
CAS04110
CAS04120
CAS04130
CAS04140
CAS04150
CAS04160
CAS04170
CAS04180

```

```

        ERROR='YES'
        WRITE(6,400)
        GO TO 36
    END IF
    IF (IDARC.GT.DAFMO(IMOFC)) THEN
*****
        FORMAT(' THE TRANSACTION DATE ,F8.0, HAS AN ILLEGAL DAY',I3,
100      WRITE(6,34) TRANSAC(I,2), IDARC
        ERROR='YES'
        WRITE(6,400)
    END IF
*****
        CHECKS COST STATEMENT DATA.
*****
        IF (TRANSAC(I,1).EQ.3) THEN
            DO 700 J=1,NMSTL
                IF (TRANSAC(I,J+2).GT.999999.OR.TRANSAC(I,J+2).LT.-999999) THEN
300      EN
*****
350      FORMAT(' COST AND DELIVERY TRANSACTIONS MUST NOT HAVE', /
400      ' VALUES LARGER THAN SIX DIGITS. THE COST STATEMENT DATED ',F8.0,
450      ' HAS A VALUE THAT IS TOO LARGE. ')
        WRITE(6,40) TRANSAC(I,2)
        ERROR='YES'
        WRITE(6,400)
    END IF
5 ***** NEXT LINE SUMS COST DATA TO DETERMINE TOTAL COST.
        TOTCOS=TOTCOS+TRANSAC(I,J+2)
        IF (TRANSAC(I,J+2).LT.0.AND.ELEMENT(J,1).NE.3) THEN
*****
600      FORMAT(' NEGATIVE COST IS PERMITTED ONLY FOR SUBCONTRACTOR
700      ' TYPE PROGRESS PAYMENTS. THE COST STATEMENT DATED ',F8.0,
800      ' HAS A NEGATIVE VALUE AND IS NOT A SUBCONTRACTOR COST ELEMENT. ')
        WRITE(6,50) TRANSAC(I,2)
        ERROR='YES'
        WRITE(6,400)
    END IF
9 ***** NEXT LINE SUMS COST DATA BY COST ELEMENT.
        ELMENT(J,5)=ELMENT(J,5)+TRANSAC(I,J+2)
100      CONTINUE
    END IF
*****
150      CHECKS ACTUAL PROGRESS PAYMENT DATA.
*****
200      IF (TRANSAC(I,1).EQ.1) THEN
        ACFFG=I
        IF (TRANSAC(I,3).LT.0) THEN
300 *****
400      FORMAT(' AN ACTUAL PROGRESS PAYMENT MUST HAVE A VALUE , /
500      ' GREATER THAN 0. THE PROGRESS PAYMENT DATED ',F8.0, ' IS LESS THAN
600      '
        WRITE(6,60) TRANSAC(I,2)
        ERROR='YES'
        WRITE(6,400)
    END IF
        IF (TRANSAC(I,4).LT.0.OR.TRANSAC(I,4).GT.1) THEN
*****
700      FORMAT(' A PROGRESS PAYMENT RATE MUST BE BETWEEN 0 AND 1.
800      ' THE TRANSACTION DATED ',F8.0, ' VIOLATES THIS RULE. ')

```

CAS04190  
 CAS04200  
 CAS04210  
 CAS04220  
 CAS04230  
 CAS04240  
 CAS04250  
 CAS04260  
 CAS04270  
 CAS04280  
 CAS04290  
 CAS04300  
 CAS04310  
 CAS04320  
 CAS04330  
 CAS04340  
 CAS04350  
 CAS04360  
 CAS04370  
 CAS04380  
 CAS04390  
 CAS04400  
 CAS04410  
 CAS04420  
 CAS04430  
 CAS04440  
 CAS04450  
 CAS04460  
 CAS04470  
 CAS04480  
 CAS04490  
 CAS04500  
 CAS04510  
 CAS04520  
 CAS04530  
 CAS04540  
 CAS04550  
 CAS04560  
 CAS04570  
 CAS04580  
 CAS04590  
 CAS04600  
 CAS04610  
 CAS04620  
 CAS04630  
 CAS04640  
 CAS04650  
 CAS04660  
 CAS04670  
 CAS04680  
 CAS04690  
 CAS04700  
 CAS04710  
 CAS04720  
 CAS04730  
 CAS04740  
 CAS04750  
 CAS04760  
 CAS04770  
 CAS04780  
 CAS04790  
 CAS04800

```

WRITE(S,70) TRNSAC(I,2)
ERROR= YES
WRITE(S,400)
END IF
END IF
IF (TRNSAC(I,1).EQ.2) THEN
*****
FORMAT(' DO NOT INPUT ANY PLANNED PROGRESS PAYMENTS INTO THE ,CAS04880
, DATA FILE. THE TRANSACTION DATED ,F8.0, MUST BE REMOVED. ')CAS04870
WRITE(S,75) TRNSAC(I,2)
ERROR= YES
WRITE(S,400)
END IF
IF (I.GT.1.AND.TRNSAC(I,2).LT.TRNSAC(I-1,2)) THEN
*****
30 FORMAT(' THE DATA IS NOT IN CHRONOLOGICAL ORDER. CHECK TRNSAC04940
TRANCTIONS , / , DATED ,F8.0, AND ,F8.0, TO CORRECT. ')CAS04930
WRITE(S,80) TRNSAC(I,2), TRNSAC(I-1,2)
ERROR= YES
WRITE(S,400)
END IF
IF (TRNSAC(I,1).EQ.4.OR.TRNSAC(I,1).EQ.5) THEN
*****
NEXT LINE SUMS DELIVERY PRICES TO DETERMINE TOTAL PRICE.
TOTFYD=TOTFYD+TRNSAC(I,3)
IF (TRNSAC(I,3).GT.999999.99.OR.TRNSAC(I,3).LE.0) THEN
*****
40 FORMAT(' A DELIVERY PAYMENT MUST HAVE A VALUE LARGER THAN ,CAS05000
5.0 BUT NOT CONTAIN MORE THAN 6 DIGITS. THE TRANSACTION DATED ,CAS04950
1.0, ,F8.0, VIOLATES THIS RULE. ')
WRITE(S,90) TRNSAC(I,2)
ERROR= YES
WRITE(S,400)
END IF
IF (TRNSAC(I,1).EQ.5) THEN
IF (TRNSAC(I,4).LT.0.OR.TRNSAC(I,4).GT.1) THEN
*****
50 FORMAT(' THE LIQUIDATION RATE ASSOCIATED WITH AN ACTUAL ,CAS05150
8. DELIVERY MUST BE BETWEEN 0 AND 1. THE TRANSACTION DATED ,F8.0,CAS05160
% VIOLATES THIS RULE. ')
WRITE(S,100) TRNSAC(I,2)
ERROR= YES
WRITE(S,400)
END IF
END IF
IF (TRNSAC(I,1).LT.1.OR.TRNSAC(I,1).GT.5) THEN
*****
110 FORMAT(' THE TRANSACTION DATED ,F8.0, HAS AN ILLEGAL , / , CAS05260
% TRANSACTION CODE. THE CODE MUST BE A 1, 2, 3, 4, OR 5. ')
WRITE(S,110) TRNSAC(I,2)
ERROR= YES
WRITE(S,400)
END IF
COSTFD=TOTFYD / (1+FRFRAT)-1
TH=TOTCDS*1.0005
IF (COSTFD.GT.TH) THEN
*****
20 FORMAT(' COST REIMBURSEMENT CANNOT BE LATER THAN INCURRED ,CAS05380
2. / , COST. THE TRANSACTION DATED ,F8.0, VIOLATES THIS RULE. ')CAS05370
FORMAT(' THE DELIVERY PAYMENTS TO DATE EXCEED THE COST TO DATE. ')CAS05360

```

```

16 FORMAT(' TOTAL COST AT THIS POINT FROM COST STATEMENTS ',I62,F8.0) CAS05400
17 FORMAT(' TOTAL DELIVERIES AT THIS POINT ',I62,F8.0) CAS05410
18 FORMAT(' COST REIMBURSED AT THIS POINT BASED ON DELIVERY PRICES ',I62,F8.0) CAS05410
19 WRITE(6,120) 'GENERAL(I,2)' CAS05420
20 WRITE(6,125) CAS05430
21 WRITE(6,126) 'TOTALS' CAS05460
22 WRITE(6,127) 'TOTALYD' CAS05470
23 WRITE(6,129) 'COSTED' CAS05480
24 ERROR= YES CAS05490
25 WRITE(6,400) CAS05500
26 END IF CAS05510
27 CONTINUE CAS05520
28 DO 500 I=1,NMESTL CAS05530
29 IF (COSTEL(I,2).NE. INCDR .AND. COSTEL(I,2).NE. PAIDC .AND. COSTEL(I,2).NE. SUBCT) THEN CAS05540
30 ***** CAS05550
31 FORMAT(' THE COST ELEMENT NUMBER ',I3, ' DOES NOT HAVE A ',I3, ' CAS05570
32 VALID COST ELEMENT CODE. ') CAS05580
33 WRITE(6,130) I CAS05590
34 ERROR= YES CAS05600
35 WRITE(6,400) CAS05610
36 END IF CAS05620
37 ***** CAS05630
38 CHECKS PAID COST AND SUBCONTRACTOR PROGRESS PAYMENT COST CAS05640
39 ELEMENT(I,1) CAS05650
40 ***** CAS05660
41 IF (ELEMENT(I,1).LT.1) THEN CAS05670
42 IF (ELEMENT(I,4).GT.100) THEN CAS05680
43 ***** CAS05690
44 FORMAT(' THE FLOAT DAYS ON COSTS PAID IN THE PRESENT',I3, ' CAS05700
45 ACCOUNTING PERIOD SHOULD BE NEGATIVE. THIS IS NOT THE CASE FOR ',I3, ' CAS05710
46 COST ELEMENT ',I3, ' ') CAS05720
47 WRITE(6,140) I CAS05730
48 ERROR= YES CAS05740
49 WRITE(6,400) CAS05750
50 END IF CAS05760
51 IF (ELEMENT(I,2).GT.28) THEN CAS05770
52 ***** CAS05780
53 FORMAT(' THE FLOAT DAYS FOR UNPAID COSTS CANNOT EXCEED ',I3, ' CAS05790
54 28 DAYS. THIS IS NOT THE CASE FOR COST ELEMENT ',I3, ' ') CAS05800
55 WRITE(6,145) I CAS05810
56 ERROR= YES CAS05820
57 WRITE(6,400) CAS05830
58 END IF CAS05840
59 IF (ELEMENT(I,3).LT.0.OR.ELEMENT(I,3).GT.1) THEN CAS05850
60 ***** CAS05860
61 FORMAT(' THE PERCENT PAID IN THE PRESENT ACCOUNTING PERIOD ',I3, ' CAS05870
62 %, ',I3, ' SHOULD BE BETWEEN 0 AND 1 FOR COST ELEMENT ',I3, ' ') CAS05880
63 WRITE(6,150) I CAS05890
64 ERROR= YES CAS05900
65 WRITE(6,400) CAS05910
66 END IF CAS05920
67 IF (ELEMENT(I,1).EQ.3.AND.ELEMENT(I,5).NE.0) THEN CAS05930
68 ***** CAS05940
69 FORMAT(' THE SUBCONTRACTOR PROGRESS PAYMENTS SHOULD ADD ',I3, ' CAS05950
70 1 TO 5 ACROSS THE LIFE OF THE CONTRACT FOR COST ELEMENT ',I3, ' ') CAS05960
71 WRITE(6,160) I CAS05970
72 ERROR= YES CAS05980

```



	WRITE(6,400)	CAS06040
	END IF	CAS06041
	END IF	CAS06042
100	CONTINUE	CAS06043
	DO 150 I=1,NCFRG	CAS06044
	IF (TRANSAC(I,1).EQ.1) THEN	CAS06045
	CACFRG=CACFRG+1	CAS06046
	END IF	CAS06047
	IF (TRANSAC(I,1).EQ.3) THEN	CAS06048
	CCOST=CCOST+1	CAS06049
	END IF	CAS06050
150	CONTINUE	CAS06051
	CCOST=CCOST-CACFRG	CAS06052
	TOTPROF=TOTPYD-TOTCOS	CAS06053
	PROFIT=TOTPYD / TOTCOS-1	CAS06054
	TODAYS=INT(TRANSAC(NMTRAN,13))	CAS06055
	LASTDEL=INT(TRANSAC(NMTRAN,2))	CAS06056
	IF (TRANSAC(1,1).NE.3) THEN	CAS06057
170	*****	CAS06058
	FORMAT(' THE FIRST TRANSACTION MUST BE A COST REPORT.')	CAS06059
	WRITE(6,170)	CAS06060
	ERROR='YES'	CAS06061
	WRITE(6,400)	CAS06062
	END IF	CAS06063
	IF (TRANSAC(NMTRAN,1).NE.4.AND. TRANSAC(NMTRAN,1).NE.5) THEN	CAS06064
180	*****	CAS06065
	FORMAT(' THE LAST TRANSACTION MUST BE A DELIVERY.')	CAS06066
	WRITE(6,180)	CAS06067
	ERROR='YES'	CAS06068
	WRITE(6,400)	CAS06069
	END IF	CAS06070
	IF (CCOST.LT.0) THEN	CAS06071
190	FORMAT(' THE DATA FILE CONTAINS MORE ACTUAL PROGRESS PAYMENTS',/	CAS06072
	2, ' THAN COST STATEMENTS. THE NUMBER OF ACTUAL PROGRESS PAYMENTS',/	CAS06073
	2, ' SHOULD EQUAL THE NUMBER OF COST STATEMENTS BEFORE THE LAST ACTUAL	CAS06074
	COST PROGRESS PAYMENT.')	CAS06075
	WRITE(6,190)	CAS06076
	ERROR='YES'	CAS06077
	WRITE(6,400)	CAS06078
	END IF	CAS06079
	FRFRERAT+0.0005	CAS06080
	FLFRFRERAT+0.0005	CAS06081
	IF (PROFIT.GT.PL.OR.PROFIT.LT.PL) THEN	CAS06082
200	*****	CAS06083
	FORMAT(' THE PROFIT RATE COMPUTED FROM THE DATA FILE (' ,FS.2, ')	CAS06084
	DOES NOT MATCH THE PROFIT',/, ' RATE (' ,FS.2, ') IN THE FIRST LINE	CAS06085
	OF THE DATA FILE.')	CAS06086
210	FORMAT(' ',T6,'COMPUTED PROFIT BASED ON THE FOLLOWING VALUES COMPUTED	CAS06087
	FROM DATA INPUTS:')	CAS06088
220	FORMAT(' ',T11,'TOTAL PRICE FROM DELIVERY PRICES ',T48,I10,/,'T11,	CAS06089
	2,' TOTAL COST FROM COST STATEMENTS ',T48,I10,/,'T11, PROFIT DOLLARS ',	CAS06090
	2,T48,I10/	CAS06091
	WRITE(6,200) PROFIT*100,FRFRERAT*100	CAS06092
	WRITE(6,210)	CAS06093
	WRITE(6,220) INT(TOTPYD), INT(TOTCOS), INT(TOTPROF)	CAS06094
	ERROR='YES'	CAS06095
	WRITE(6,400)	CAS06096
	END IF	CAS06097
	*****	CAS06098
	SUMS COST STATEMENT DATA WHEN ERRORS ARE DETECTED IN	CAS06099

```

      EDIT SUBROUTINE.
      *****
      IF (ERROR.EQ.'YES') THEN
        DO 300 I=1,NMTRAN
          IF (TRANSAC(I,1).EQ.3) THEN
            DO 310 J=1,NMSTL
              TRANSAC(I,14)=TRANSAC(I,14)+TRANSAC(I,J+2)
            CONTINUE
          END IF
        CONTINUE
      END IF
      300  FORMAT( / )
      RETURN
      END
      *****      END OF SUBROUTINE EDIT      *****

      *****      START OF SUBROUTINE DATAIN      *****

      SUBROUTINE DATAIN(FRGLAG,DELLAG,FLXFLG,FRGRTE,FRGFLG,
      3ALTLDY,TRNFLG,CSTFLG,RERUN,TWOFLG)
      INTEGER FRGLAG,DELLAG,FRGFLG,ALTLDY,TWOFLG
      CHARACTER*3 TRNFLG,CSTFLG,FLXFLG
      REAL FRGRTE
      30  FORMAT(' THE FOLLOWING DATA IS REQUESTED OF THE USER AT THE ')
      31  FORMAT(' TERMINAL SO DIFFERENT CALCULATIONS CAN BE MADE USING ')
      32  FORMAT(' THE SAME DATA FILE.')
      33  FORMAT(' WHAT IS THE PROGRESS PAYMENT LAG TIME IN DAYS? (TIME )')
      34  FORMAT(' FROM WHEN STATEMENT IS ISSUED UNTIL PAYMENT IS REC-')
      35  FORMAT(' EIVED? )')
      36  FORMAT(' WHAT IS THE DELIVERY PAYMENT LAG TIME IN DAYS? (TIME )')
      37  FORMAT(' FROM WHEN DELIVERY IS MADE UNTIL PAYMENT IS RECEIVED. )')
      38  FORMAT(' WOULD YOU LIKE TO SPECIFY A CERTAIN PROGRESS PAYMENT?')
      39  FORMAT(' RATE FOR THESE CALCULATIONS? A NO ANSWER IMPLIES YOU')
      40  FORMAT(' WANT THE COMPUTER TO CALCULATE THE FLEXIBLE PROGRESS')
      41  FORMAT(' RATE USING THE CONTRACTOR INVESTMENT RATIO. (YES / NO)')
      42  FORMAT(' WHAT PROGRESS PAYMENT RATE WOULD YOU LIKE TO SPECIFY?')
      43  FORMAT(' (PLEASE USE THE FORMAT 0.XX) )')
      44  FORMAT(' AFTER HOW MANY DAYS WOULD YOU LIKE THE ALTERNATE')
      45  FORMAT(' LIQUIDATION RATE USED IN CALCULATIONS? VALUE MUST BE')
      46  FORMAT(' GREATER THAN OR EQUAL TO 365. )')
      47  FORMAT(' WOULD YOU LIKE TO SEE A TRANSACTION REPORT? (YES / NO)')
      48  FORMAT(' WOULD YOU LIKE TO SEE A RECAP OF THE COST REPORT DATA?')
      49  FORMAT(' (YES / NO) )')
      50  FORMAT(' THE PROGRESS PAYMENT LAG MUST BE POSITIVE AND CANNOT EXCEED 56 DAYS. )')
      51  FORMAT(' THE DELIVERY PAYMENT LAG MUST BE POSITIVE AND CANNOT EXCEED 60 DAYS. )')
      52  FORMAT( / )
      53  FORMAT( / )
      54  FORMAT(' THE ALTERNATE LIQUIDATION RATE CANNOT BE USED UNTIL )')
      55  FORMAT(' ONE YEAR HAS ELAPSED ON THE CONTRACT. (365 DAYS) )')
      IF (RERUN.EQ.0) THEN
        WRITE(6,30)
        WRITE(6,31)
        WRITE(6,32)
      END IF
      *****
      ASKS FOR PROGRESS PAYMENT LAG.

```

CAS06590  
 CAS06600  
 CAS06610  
 CAS06620  
 CAS06630  
 CAS06640  
 CAS06650  
 CAS06660  
 CAS06670  
 CAS06680  
 CAS06690  
 CAS06700  
 CAS06710  
 CAS06720  
 CAS06730  
 CAS06740  
 CAS06750  
 CAS06760  
 CAS06770  
 CAS06780  
 CAS06790  
 CAS06800  
 CAS06810  
 CAS06820  
 CAS06830  
 CAS06840  
 CAS06850  
 CAS06860  
 CAS06870  
 CAS06880  
 CAS06890  
 CAS06900  
 CAS06910  
 CAS06920  
 CAS06930  
 CAS06940  
 CAS06950  
 CAS06960  
 CAS06970  
 CAS06980  
 CAS06990  
 CAS07000  
 CAS07010  
 CAS07020  
 CAS07030  
 CAS07040  
 CAS07050  
 CAS07060  
 CAS07070  
 CAS07080  
 CAS07090  
 CAS07100  
 CAS07110  
 CAS07120  
 CAS07130  
 CAS07140  
 CAS07150  
 CAS07160  
 CAS07170  
 CAS07180

	*****	CAS07190
WRITE(6,300)		CAS07200
WRITE(6,60)		CAS07210
WRITE(6,61)		CAS07220
WRITE(6,62)		CAS07230
READ(5,FMT=*) , PRGLAG		CAS07240
IF (PRGLAG.LT.0.OR.PRGLAG.GT.56) THEN		CAS07250
WRITE(6,250)		CAS07260
GO TO 340		CAS07270
END IF		CAS07280
*****		CAS07290
HIGHLIGHTS TWO COST STATEMENTS MAY OCCUR BETWEEN PROGRESS		CAS07300
PAYMENTS.		CAS07310
*****		CAS07320
IF (PRGLAG.GE.28) THEN		CAS07330
TWOFLG=1		CAS07340
ELSE		CAS07350
TWOFLG=0		CAS07360
END IF		CAS07370
*****		CAS07380
ADDS FOR DELIVERY PAYMENT LAG.		CAS07390
*****		CAS07400
410 WRITE(6,290)		CAS07410
WRITE(6,80)		CAS07420
WRITE(6,81)		CAS07430
READ(5,FMT=*) , DELLAG		CAS07440
IF (DELLAG.LT.0.OR.DELLAG.GT.60) THEN		CAS07450
WRITE(6,270)		CAS07460
GO TO 410		CAS07470
END IF		CAS07480
*****		CAS07490
ASKS WHETHER FLEX RATE IS DESIRED.		CAS07500
*****		CAS07510
480 WRITE(6,290)		CAS07520
WRITE(6,130)		CAS07530
WRITE(6,131)		CAS07540
WRITE(6,132)		CAS07550
WRITE(6,133)		CAS07560
READ(5,FMT='(A3)') FLXFLG		CAS07570
IF (FLXFLG(1:1).NE.'Y'.AND.FLXFLG(1:1).NE.'N') THEN		CAS07580
GO TO 480		CAS07590
END IF		CAS07600
*****		CAS07610
ALLOWS SPECIFIC PROGRESS PAYMENT RATE TO BE USED IN		CAS07620
LIEU OF FLEX RATE.		CAS07630
*****		CAS07640
IF (FLXFLG(1:1).EQ.'Y') THEN		CAS07650
530 WRITE(6,290)		CAS07660
WRITE(6,170)		CAS07670
WRITE(6,171)		CAS07680
READ(5,FMT=*) , PRGRTE		CAS07690
IF (PRGRTE.LT.0.OR.PRGRTE.GT.1) THEN		CAS07700
550 FORMAT(' THE PROGRESS PAYMENT RATE MUST BE BETWEEN 1 AND 0.		CAS07710
570 WRITE(6,570)		CAS07720
GO TO 530		CAS07730
END IF		CAS07740
PRDFLG=0		CAS07750
ELSE		CAS07760
PRGRTE=1.0		CAS07780

	FRGFLG=1	CAS07790
	END IF	CAS07800
	*****	CAS07810
	ASKS WHEN TO SWITCH TO ALTERNATE LIQUIDATION RATE.	CAS07820
	*****	CAS07830
	WRITE(6,290)	CAS07840
	WRITE(6,190)	CAS07850
	WRITE(6,191)	CAS07860
	WRITE(6,192) p v	CAS07870
	READ(5,FMT=*), ALTLDY	CAS07880
	IF (ALTLDY.LT.365) THEN	CAS07890
	WRITE(6,310)	CAS07900
	WRITE(6,311)	CAS07910
	GO TO 750	CAS07920
	END IF	CAS07930
	*****	CAS07940
	ASKS IF TRANSACTION REPORT IS DESIRED.	CAS07950
	*****	CAS07960
820	WRITE(6,290)	CAS07970
	WRITE(6,220)	CAS07980
	READ(5,FMT=(A3)) TRNFLG	CAS07990
	IF (TRNFLG(1:1).NE.'Y'.AND.TRNFLG(1:1).NE.'N') THEN	CAS08000
	GO TO 820	CAS08010
	END IF	CAS08020
	*****	CAS08030
	ASKS IF COST RECAP REPORT IS DESIRED.	CAS08040
	*****	CAS08050
850	WRITE(6,290)	CAS08060
	WRITE(6,230)	CAS08070
	READ(5,FMT=(A3)) CSTFLG	CAS08080
	IF (CSTFLG(1:1).NE.'Y'.AND.CSTFLG(1:1).NE.'N') THEN	CAS08090
	GO TO 860	CAS08100
	END IF	CAS08110
	WRITE(6,300)	CAS08120
	RETURN	CAS08130
	END	CAS08140
	*****          END OF SUBROUTINE DATAIN          *****	CAS08150
		CAS08160
		CAS08170
		CAS08180
	*****          START OF SUBROUTINE CREFAY          *****	CAS08190
		CAS08200
	SUBROUTINE CREFAY(TRANSAC,NMTRAN,ACPRG,CCOST,FRGLAG,FRGFLG,STDAY,	CAS08210
	&DELLAG,ERROR)	CAS08220
	REAL TRANSAC(325,20),CCOST	CAS08230
	INTEGER NMTRAN,ACPRG,FRGLAG,FRGFLG,STDAY,DELLAG,Z	CAS08240
	INTEGER CNTFLG,FRG1,DAYS	CAS08250
	CHARACTER*3 ERROR	CAS08260
	FRG1=0	CAS08270
	FRGFLG=0	CAS08280
	*****	CAS08290
	USED TO MATCH COST STATEMENTS TO PROGRESS PAYMENTS	CAS08300
	WHEN TWO COST STATEMENTS APPEAR BETWEEN ACTUAL PROGRESS	CAS08310
	PAYMENTS.	CAS08320
	*****	CAS08330
	IF (CCOST.GT.0) THEN	CAS08340
	CNTFLG=0	CAS08350
	DO 100 I=ACPRG,1,-1	CAS08360
	IF (TRANSAC(I,1).EQ.3.AND.CNTFLG.EQ.0) THEN	CAS08370
	CCOST=CCOST-1	CAS08380

```

        ACFRG=1
        CNTFLG=1
        END IF
100    CONTINUE
        END IF
        *****
        ADDS DELIVERY LAG TO EACH DELIVERY DATE INPUT TO
        ESTABLISH DATE ON WHICH PAYMENT IS RECEIVED.
        *****
DO 150 I = 1,NMTRAN
    IF (TRANSAC(I,1).EQ.4 .OR. TRANSAC(I,1).EQ.5) THEN
        TRANSAC(I,13)=TRANSAC(I,13)+DELLAG
        K=INT(TRANSAC(I,13))+STDAY
        Z=K
        TRANSAC(I,2)=KDATE(K)
    END IF
150 CONTINUE
        *****
        CREATES A PLANNED PROGRESS PAYMENT TRANSACTION FOR EACH
        COST STATEMENT NOT COVERED BY AN ACTUAL PROGRESS PAYMENT.
        *****
DO 200 I=ACFRG,NMTRAN
    IF (TRANSAC(I,1).EQ.3.AND.((INT(TRANSAC(I,13))+PRGLAG+STDAY).LT.Z) /
& THEN
        PROG1=PROG1+1
        NMTRAN=NMTRAN+1
        TRANSAC(NMTRAN,1)=2
        TRANSAC(NMTRAN,13)=TRANSAC(I,13)+PRGLAG
        K=INT(TRANSAC(NMTRAN,13))+STDAY
        TRANSAC(NMTRAN,2)=KDATE(K)
        PRGFLG=1
    ELSE
        GO TO 200
    END IF
        *****
        CHECKS TO SEE IF PLANNED PROGRESS PAYMENTS ARE AT LEAST
        28 DAYS APART.  MAKES SURE PROGRESS PAYMENTS ARE BILLED
        ONLY ONCE PER MONTH.
        *****
    IF (PROG1.GT.1) THEN
        DAYS=TRANSAC(NMTRAN,13)-TRANSAC(NMTRAN-1,13)
    END IF
    IF (DAYS.LT.28.AND.PROG1.GT.1) THEN
        WRITE(6,300) TRANSAC(NMTRAN,2)
        ERROR='YES'
        WRITE(6,400)
    END IF
200 CONTINUE
C ***** NEXT LINE SORTS TRANSACTIONS AFTER ALL NEW TRANSACTIONS
C ***** ARE CREATED.
    CALL SORT(TRANSAC,NMTRAN,20,13)
100 FORMAT(' PLANNED PROGRESS PAYMENTS CANNOT OCCUR LESS',/,' THAN 28 DAYS',/,'
& DAYS APART.  THE TRANSACTION DATED',/,'F7.0,',/,' VIOLATES THIS RULE.  ',/,'
& CHECK INPUT FILE',/,' TO INSURE COST STATEMENTS ARE AT LEAST 28 DAYS',/,'
& ',/,' APART. ')
100 FORMAT(/)
RETURN
END
        *****          END OF SUBROUTINE CREFAY          *****

```

		CAS08990
		CAS09000
	***** START OF SUBROUTINE ACTFPY *****	CAS09010
		CAS09020
	SUBROUTINE ACTFPY(TRANSAC,A,OUTPUT,FLOATD,CSTFLT,PRFRAT,ALTLDY,STCOST,PPCNT)	CAS09030
	REAL TRANSAC(325,20),OUTPUT(325,10),CSTFLT(1000,2),PRFRAT,STCOST	CAS09040
	INTEGER A,FLOATD,ALTLDY,PPCNT	CAS09050
	*****	CAS09060
	DETERMINES PAID INCURRED COST AT THE TIME OF THE	CAS09070
	TRANSACTION.	CAS09080
	*****	CAS09090
	DO 80 I=1,FLOATD	CAS09100
	IF (CSTFLT(I,1).LE.TRNSAC(A,13)) THEN	CAS09110
	TRANSAC(A,17)=CSTFLT(I,2)+TRANSAC(A,17)	CAS09120
	END IF	CAS09130
80	CONTINUE	CAS09140
	TRANSAC(A,17)=TRANSAC(A,17)+STCOST	CAS09150
	*****	CAS09160
	NEXT THREE LINES DO THE FOLLOWING: ROLL COST ELIGIBLE	CAS09170
	FOR PROGRESS PAYMENT FORWARD; ROLL COST ELIGIBLE FOR 100 0/100	CAS09180
	PROGRESS PAYMENT FORWARD; AND, KEEP TRACK OF THE CURRENT	CAS09190
	PROGRESS PAYMENT RATE.	CAS09200
	*****	CAS09210
	TRANSAC(A,15)=TRANSAC(A-1,15)	CAS09220
	TRANSAC(A,16)=TRANSAC(A-1,16)	CAS09230
	TRANSAC(A,18)=TRANSAC(A,4)	CAS09240
	*****	CAS09250
	KEEPS TRACK OF LIQUIDATION RATE.	CAS09260
	*****	CAS09270
	IF (TRANSAC(A,13).LT.ALTLDY) THEN	CAS09280
	TRANSAC(A,19)=TRANSAC(A,4)	CAS09290
	ELSE	CAS09300
	TRANSAC(A,19)=TRANSAC(A,4) / (1+PRFRAT)	CAS09310
	END IF	CAS09320
	*****	CAS09330
	FILLS IN OUTPUT FOR TRANSACTION REPORT.	CAS09340
	*****	CAS09350
	OUTPUT(A,1)=OUTPUT(A-1,1)	CAS09360
	OUTPUT(A,2)=OUTPUT(A,1)-TRANSAC(A,17)	CAS09370
	OUTPUT(A,3)=TRANSAC(A,17)	CAS09380
	OUTPUT(A,4)=OUTPUT(A-1,4)+TRANSAC(A,3)	CAS09390
	OUTPUT(A,5)=OUTPUT(A-1,5)	CAS09400
	OUTPUT(A,6)=OUTPUT(A,4)+OUTPUT(A,5)	CAS09410
	OUTPUT(A,7)=OUTPUT(A-1,7)	CAS09420
	OUTPUT(A,8)=OUTPUT(A,6)+OUTPUT(A,7)	CAS09430
	OUTPUT(A,9)=OUTPUT(A,3)-OUTPUT(A,6)	CAS09440
	OUTPUT(A,10)=OUTPUT(A-1,10)+TRANSAC(A,3)	CAS09450
		CAS09460
	***** NEXT LINE IS A COUNTER FOR PROGRESS PAYMENTS. USED IN	CAS09470
	***** SUBROUTINE PRGPAY IF PROGRESS PAYMENT LAG EXCEEDS COST	CAS09480
	***** STATEMENT PERIOD. USED TO MATCH PROGRESS PAYMENTS WITH	CAS09490
	***** PROPER COST STATEMENT DATA.	CAS09500
	PPCNT=PPCNT+1	CAS09510
	RETURN	CAS09520
	END	CAS09530
	***** END OF SUBROUTINE ACTFPY *****	CAS09540
		CAS09550
		CAS09560
		CAS09570
	***** START OF SUBROUTINE PRGPAY *****	CAS09580

	SUBROUTINE PRGPAY (TRNSAC,A,FRGRTE,CSTFLT,FLOATD,OUTPUT,TOTCHG,	CAS09690
	SUMCST,TOTPYD,ALTLDY,FRFRAT,STCOST,PPCNT,CSTDAY,TWOFLG)	CAS09691
	REAL FRM15, FRM16, FRM19, FRM20E, FRM21E, FRM25,PAYMNT	CAS09692
	REAL TRNSAC (25,20),FRGRTE,CSTFLT (1000,2),OUTPUT (25,10),TOTCHG,	CAS09693
	SUMCST,FRFRAT,TOTPYD,STCOST	CAS09694
	INTEGER A,FLOATD,ALTLDY,PPCNT,CSTDAY (170),TWOFLG	CAS09695
C	***** COUNTER DESCRIBED ABOVE AT END OF SUBROUTINE ACTPPY.	CAS09696
	PPCNT=PPCNT+1	CAS09697
C	***** NEXT 14 LINES DO SAME THING AS DESCRIBED ABOVE IN SUBROUTINE	CAS09698
C	***** ACTPPY.	CAS09699
	TRNSAC (A,15)=TRNSAC (A-1,15)	CAS09700
	TRNSAC (A,16)=TRNSAC (A-1,16)	CAS09710
	DO 100 I=1,FLOATD	CAS09720
	IF (CSTFLT (I,1).LE.TRNSAC (A,13)) THEN	CAS09730
	TRNSAC (A,17)=TRNSAC (A,17)+CSTFLT (I,2)	CAS09740
	END IF	CAS09750
100	CONTINUE	CAS09760
	TRNSAC (A,17)=TRNSAC (A,17)+STCOST	CAS09770
	TRNSAC (A,18)=FRGRTE	CAS09780
	IF (TRNSAC (A,13).LT.ALTLDY) THEN	CAS09790
	TRNSAC (A,19)=TRNSAC (A,18)	CAS09800
	ELSE	CAS09810
	TRNSAC (A,19)=TRNSAC (A,18) / (1+FRFRAT)	CAS09820
	END IF	CAS09830
C	*****	CAS09840
C	MATCHES PROGRESS PAYMENT TO PROPER COST STATEMENT DATA	CAS09850
C	WHEN PROGRESS PAYMENT LAG EXCEEDS COST STATEMENT PERIOD.	CAS09860
C		CAS09870
C	NOTE: ALL VARIABLES FRMAX EQUATE TO LINE NUMBERS ON	CAS09880
C	DD FORM 1195 (PROGRESS PAYMENT FORM).	CAS09890
C	*****	CAS09900
	IF (TWOFLG.EQ.1) THEN	CAS09910
	FRM15=TRNSAC (CSTDAY (PPCNT),15)*TRNSAC (A,18)+TRNSAC (CSTDAY (PPCNT)	CAS09920
	,16)	CAS09930
	GO TO 110	CAS09940
	END IF	CAS09950
	FRM15=TRNSAC (A,15)*TRNSAC (A,18)+TRNSAC (A,16)	CAS09960
110	FRM16=TOTPYD*TRNSAC (A,19)	CAS09970
	IF (FRM16.LT.FR15) THEN	CAS09980
	FRM15=FRM16	CAS09990
	END IF	CAS10000
	FRM19=FRM15-OUTPUT (A-1,4)	CAS10010
	IF (TWOFLG.EQ.1) THEN	CAS10020
	FRM20E=(TRNSAC (CSTDAY (PPCNT),15)-SUMCST)*TRNSAC (A,18)+TRNSAC (CSTDAY	CAS10030
	PAY (PPCNT),16)	CAS10040
	GO TO 120	CAS10050
	END IF	CAS10060
	FRM20E=(TRNSAC (A,15)-SUMCST)*TRNSAC (A,18)+TRNSAC (A,16)	CAS10070
120	FRM21E=(TOTPYD-TOTCHG)*TRNSAC (A,19)	CAS10080
	IF (FRM21E.LT.FR20E) THEN	CAS10090
	FRM20E=FRM21E	CAS10100
	END IF	CAS10110
	FRM25=FRM20E-OUTPUT (A-1,10)	CAS10120
	IF (FRM25.LT.FR19) THEN	CAS10130
	FRM19=FRM25	CAS10140
	END IF	CAS10150
	PAYMNT=FRM19	CAS10160
C	*****	CAS10170
C	FILLS IN OUTPUT FOR TRANSACTION REPORT.	CAS10180

```

*****
OUTPUT(A,1)=OUTPUT(A-1,1)
OUTPUT(A,2)=OUTPUT(A,1)-TRANSAC(A,17)
OUTPUT(A,3)=TRANSAC(A,17)
OUTPUT(A,4)=OUTPUT(A-1,4)+PAYMNT
OUTPUT(A,5)=OUTPUT(A-1,5)
OUTPUT(A,6)=OUTPUT(A,4)+OUTPUT(A,5)
OUTPUT(A,7)=OUTPUT(A-1,7)
OUTPUT(A,8)=OUTPUT(A,6)+OUTPUT(A,7)
OUTPUT(A,9)=OUTPUT(A,3)-OUTPUT(A,6)
OUTPUT(A,10)=OUTPUT(A-1,10)+PAYMNT
RETURN
END

```

```

*****      END OF SUBROUTINE PRGPAY      *****

```

```

*****      START OF SUBROUTINE COSTST      B*****

```

```

SUBROUTINE COSTST(TRANSAC,A,NMCSTL,ELEMENT,FLOATD,CSTFLT,OUTPUT,
&SUBCNT,UPDCNT,UPDFLT,UPDSUB,PRGRTE,STCOST,CNTST,CSTDAY,TWOFLG)
REAL TRANSAC(325,20),ELEMENT(10,5),CSTFLT(1000,2),OUTPUT(325,10),
&UPDFLT(1000,2),UPDSUB(1000,2),PRGRTE,STCOST
INTEGER A,NMCSTL,FLOATD,SUBCNT,UPDCNT,CNTST,CSTDAY(170),TWOFLG
REAL PAID,UNPAID

```

```

*****
      DETERMINES PAID INCURRED COST ELIGIBILITY FOR PROGRESS
      PAYMENTS.

```

```

*****
DO 180 I=1,NMCSTL
  TRANSAC(A,14)=TRANSAC(A,14)+TRANSAC(A,I+2)
  FLOATD=FLOATD+1

```

```

C ***** COST ELIGIBILITY CODE 'INCUR'
  IF (ELEMENT(I,1).EQ.1) THEN
    CSTFLT(FLOATD,1)=TRANSAC(A,13)+ELEMENT(I,2)
    CSTFLT(FLOATD,2)=TRANSAC(A,I+2)
    TRANSAC(A,15)=TRANSAC(A,15)+TRANSAC(A,I+2)
  END IF

```

```

C ***** COST ELIGIBILITY CODE 'PAIDC'
  IF (ELEMENT(I,1).EQ.2) THEN
    PAID=TRANSAC(A,I+2)*ELEMENT(I,3)
    UNPAID=TRANSAC(A,I+2)*(1-ELEMENT(I,3))
    CSTFLT(FLOATD,1)=TRANSAC(A,13)+ELEMENT(I,2)
    CSTFLT(FLOATD,2)=UNPAID
    UPDCNT=UPDCNT+1
    UPDFLT(UPDCNT,1)=CSTFLT(FLOATD,1)
    UPDFLT(UPDCNT,2)=UNPAID
    FLOATD=FLOATD+1
    CSTFLT(FLOATD,1)=TRANSAC(A,13)+ELEMENT(I,4)
    CSTFLT(FLOATD,2)=PAID
    TRANSAC(A,15)=TRANSAC(A,15)+PAID
  END IF

```

```

C ***** COST ELIGIBILITY CODE 'SUBCT'
  IF (ELEMENT(I,1).EQ.3) THEN
    PAID=TRANSAC(A,I+2)*ELEMENT(I,3)
    UNPAID=TRANSAC(A,I+2)*(1-ELEMENT(I,3))
    CSTFLT(FLOATD,1)=TRANSAC(A,13)+ELEMENT(I,2)
    CSTFLT(FLOATD,2)=UNPAID
    SUBCNT=SUBCNT+1
    UPDSUB(SUBCNT,1)=CSTFLT(FLOATD,1)

```

CAS10190  
 CAS10200  
 CAS10210  
 CAS10220  
 CAS10230  
 CAS10240  
 CAS10250  
 CAS10260  
 CAS10270  
 CAS10280  
 CAS10290  
 CAS10300  
 CAS10310  
 CAS10320  
 CAS10330  
 CAS10340  
 CAS10350  
 CAS10360  
 CAS10370  
 CAS10380  
 CAS10390  
 CAS10400  
 CAS10410  
 CAS10420  
 CAS10430  
 CAS10440  
 CAS10450  
 CAS10460  
 CAS10470  
 CAS10480  
 CAS10490  
 CAS10500  
 CAS10510  
 CAS10520  
 CAS10530  
 CAS10540  
 CAS10550  
 CAS10560  
 CAS10570  
 CAS10580  
 CAS10590  
 CAS10600  
 CAS10610  
 CAS10620  
 CAS10630  
 CAS10640  
 CAS10650  
 CAS10660  
 CAS10670  
 CAS10680  
 CAS10690  
 CAS10700  
 CAS10710  
 CAS10720  
 CAS10730  
 CAS10740  
 CAS10750  
 CAS10760  
 CAS10770  
 CAS10780  
 CAS10790



UPDSUB(SUBCNT,2)=UNPAID	CAS10780
FLOATD=FLOATD+1	CAS10800
CSTFLT(FLOATD,1)=TRANSAC(A,13)+ELEMENT(I,4)	CAS10810
CSTFLT(FLOATD,2)=PAID	CAS10820
TRANSAC(A,16)=TRANSAC(A,16)+PAID	CAS10830
END IF	CAS10840
380 CONTINUE	CAS10850
C *****	CAS10860
C PUTS UNPAID FLOATED COSTS FOR COST ELIGIBILITY CODE	CAS10870
C 'PAIDC' INTO COST ELIGIBLE FOR PROGRESS PAYMENTS AFTER	CAS10880
C FLOAT PERIOD EXPIRES.	CAS10890
C *****	CAS10900
DO 450 I=1,UPDCNT	CAS10910
IF (UPDFLT(I,1).LE.TRNSAC(A,13)) THEN	CAS10920
TRANSAC(A,15)=TRANSAC(A,15)+UPDFLT(I,2)	CAS10930
UPDFLT(I,1)=100000	CAS10940
UPDFLT(I,2)=0	CAS10950
END IF	CAS10960
450 CONTINUE	CAS10970
C *****	CAS10980
C PUTS UNPAID FLOATED COSTS FOR COST ELIGIBILITY CODE	CAS10990
C 'SUBCT' INTO COST ELIGIBLE FOR 100 0/0 PROGRESS PAYMENTS	CAS11000
C AFTER FLOAT PERIOD EXPIRES.	CAS11010
C *****	CAS11020
DO 520 I=1,SUBCNT	CAS11030
IF (UPDSUB(I,1).LE.TRNSAC(A,13)) THEN	CAS11040
TRANSAC(A,16)=TRANSAC(A,16)+UPDSUB(I,2)	CAS11050
UPDSUB(I,1)=100000	CAS11060
UPDSUB(I,2)=0	CAS11070
END IF	CAS11080
520 CONTINUE	CAS11090
C *****	CAS11100
C DETERMINES PAID INCURRED COST AT THE TIME OF THE	CAS11110
C TRANSACTION.	CAS11120
C *****	CAS11130
DO 570 I=1,FLOATD	CAS11140
IF (CSTFLT(I,1).LE.TRNSAC(A,13)) THEN	CAS11150
TRANSAC(A,17)=TRANSAC(A,17)+CSTFLT(I,2)	CAS11160
END IF	CAS11170
570 CONTINUE	CAS11180
TRANSAC(A,17)=TRANSAC(A,17)+STCOST	CAS11190
C *****	CAS11200
C FILLS IN OUTPUT FOR TRANSACTION REPORT. ALSO ROLLS	CAS11210
C COSTS ELIGIBLE FOR PROGRESS PAYMENTS FORWARD AND	CAS11220
C KEEPS TRACK OF PROGRESS PAYMENT AND LIQUIDATION RATES.	CAS11230
C *****	CAS11240
IF (A.GT.1) THEN	CAS11250
TRANSAC(A,15)=TRANSAC(A,15)+TRANSAC(A-1,15)	CAS11260
TRANSAC(A,16)=TRANSAC(A,16)+TRANSAC(A-1,16)	CAS11270
TRANSAC(A,18)=TRANSAC(A-1,18)	CAS11280
TRANSAC(A,19)=TRANSAC(A-1,19)	CAS11290
OUTPUT(A,1)=OUTPUT(A-1,1)+TRANSAC(A,14)	CAS11300
OUTPUT(A,4)=OUTPUT(A-1,4)	CAS11310
OUTPUT(A,5)=OUTPUT(A-1,5)	CAS11320
OUTPUT(A,6)=OUTPUT(A-1,6)	CAS11330
OUTPUT(A,7)=OUTPUT(A-1,7)	CAS11340
OUTPUT(A,8)=OUTPUT(A-1,8)	CAS11350
OUTPUT(A,10)=OUTPUT(A-1,10)	CAS11360
ELSE	CAS11370
TRANSAC(A,15)=TRANSAC(A,15)+STCOST	CAS11380

```

        OUTPUT(A,1)=TRANSAC(A,14)+STCOST
        TRANSAC(A,18)=PRGRTE
        TRANSAC(A,19)=PRGRTE
    END IF

```

```

        OUTPUT(A,2)=OUTPUT(A,1)-TRANSAC(A,17)
        OUTPUT(A,3)=TRANSAC(A,17)
        OUTPUT(A,9)=OUTPUT(A,3)-OUTPUT(A,6)

```

```

        *****
        USED WHEN PROGRESS PAYMENT LAG EXCEEDS COST STATEMENT
        PERIOD. MATCHES PROGRESS PAYMENTS WITH PROPER COST
        STATEMENT DATA.
        *****

```

```

    IF (TWOF LG.EQ.1) THEN
        CNTST=CNTST+1
        STDAY(CNTST)=A
    END IF
    RETURN
END

```

```

        *****          END OF SUBROUTINE COSTST          *****

```

```

        *****          START OF SUBROUTINE FUNDEL          *****

```

```

    SUBROUTINE FUNDEL(TRANSAC,A,ALTLDY,PRFRAT,ALOFLG,CSTFLT,FLOATD,
    TOTCHG,SUMCOST,OUTPUT,PRGRTE,LSTDEL,DELLAG,STCOST,TODAYS,STDAY,
    PRFRTE,LFLG)

```

```

        REAL PRFRAY, LIQDAT, COST, NETPAY,ALTRF

```

```

        REAL TRANSAC(100,20),PRFRAT,CSTFLT(100,2),TOTCHG,SUMCOST,OUTPUT(100,5)
        REAL STCOST,TODAYS

```

```

        INTEGER A,ALTLDY,ALOFLG,FLOATD,STDAY,DELLAG

```

```

    ***** NEXT THREE LINES PERFORM SAME OPERATIONS AS DESCRIBED IN

```

```

    ***** SUBROUTINE ACTPAY.

```

```

        TRANSAC(A,15)=TRANSAC(A-1,15)

```

```

        TRANSAC(A,16)=TRANSAC(A-1,16)

```

```

        TRANSAC(A,18)=TRANSAC(A-1,18)

```

```

    ***** NEXT LINE DETERMINES DATE OF LAST DELIVERY TRANSACTION IN

```

```

    ***** TERMS OF DAYS AFTER AWARD.

```

```

        K=INT(TODAYS+STDAY)+DELLAG

```

```

    ***** NEXT SIX LINES DETERMINE PAID INCURRED COST AT THE TIME

```

```

    ***** OF THE TRANSACTION AS PREVIOUSLY DESCRIBED.

```

```

        DO 110 I=1,FLOATD

```

```

            IF (CSTFLT(I,1).LE.TRANSAC(A,13)) THEN

```

```

                TRANSAC(A,17)=TRANSAC(A,17)+CSTFLT(I,2)

```

```

            END IF

```

```

        CONTINUE

```

```

        TRANSAC(A,17)=TRANSAC(A,17)+STCOST

```

```

        *****
        CONVERTS TO ALTERNATE LIQUIDATION RATE AND MAKES PROFIT
        CATCH-UP ADJUSTMENT.
        *****

```

```

    IF (TRANSAC(A,13).GT.ALTLDY) THEN

```

```

        TRANSAC(A,15)=TRANSAC(A,18) / (1+PRFRAT)

```

```

        CATCUP=0

```

```

        IF (ALOFLG.EQ.0) THEN

```

```

            CATCUP=ALTRF-OUTPUT(A-1,7)

```

```

            ALOFLG=1

```

```

        END IF

```

```

    ELSE

```

```

        CATCUP=0

```

CAS11129  
 CAS11140  
 CAS111410  
 CAS111420  
 CAS111430  
 CAS111440  
 CAS111450  
 CAS111460  
 CAS111470  
 CAS111480  
 CAS111490  
 CAS111500  
 CAS111510  
 CAS111520  
 CAS111530  
 CAS111540  
 CAS111550  
 CAS111560  
 CAS111570  
 CAS111580  
 CAS111590  
 CAS111600  
 CAS111610  
 CAS111620  
 CAS111630  
 CAS111640  
 CAS111650  
 CAS111660  
 CAS111670  
 CAS111680  
 CAS111690  
 CAS111700  
 CAS111710  
 CAS111720  
 CAS111730  
 CAS111740  
 CAS111750  
 CAS111760  
 CAS111770  
 CAS111780  
 CAS111790  
 CAS111800  
 CAS111810  
 CAS111820  
 CAS111830  
 CAS111840  
 CAS111850  
 CAS111860  
 CAS111870  
 CAS111880  
 CAS111890  
 CAS111900  
 CAS111910  
 CAS111920  
 CAS111930  
 CAS111940  
 CAS111950  
 CAS111960  
 CAS111970  
 CAS111980

1951 11 10

**Figure 6.** The effect of the concentration of the inhibitor on the rate of polymerization of styrene initiated by AIBN at 70°C. [Styrene] = 0.8 mol/L; [AIBN] = 0.001 mol/L; [Inhibitor] = 0–0.001 mol/L. (●) DMSO; (○) DMF; (□) NMP; (△) THF; (◇) CH<sub>2</sub>Cl<sub>2</sub>; (▽) CHCl<sub>3</sub>.

CUB11000  
 CUB11001  
 CUB11002  
 CUB11003  
 CUB11004  
 CUB11005  
 CUB11006  
 CUB11007

[illegible][illegible]

CRS 121-100  
CRS 121-150  
CRS 121-200  
CRS 121-250  
CRS 121-300  
CRS 121-350  
CRS 121-400

## References

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	5																																																																														

LMS1-17  
 LMS1-18  
 LMS1-19  
 LMS1-20

CMS124-  
CMS124-0

CAS12440  
CAS12450  
CAS12460  
CAS12470  
CAS12480  
CAS12490  
CAS12500  
CAS12510  
CAS12520  
CAS12530  
CAS12540

1. 1990年12月1日以前  
 2. 1990年12月1日以后  
 3. 1990年12月1日以前

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 84

1000

```

***** START OF SUBROUTINE ACTDEL *****
***** CALCULATIONS ARE THE SAME AS DESCRIBED IN SUBROUTINE FLNDEL
SUBROUTINE ACTDEL (TRANSAC,A,PRFRAT,ALTDY,ALQFLG,CSTFLT,FLOATD,
  TOTCHG,SUMCST,OUTPUT,LSTDEL,DELLAG,STCOST,TODAYS,STDAY,ALTERF)
  REAL TRANSAC(325,20),PRFRAT,CSTFLT(1000,2),TOTCHG,SUMCST,OUTPUT(325
  0,10),STCOST,TODAYS,ALTERF
  INTEGER A,ALTDY,ALQFLG,FLOATD,STDAY,DELLAG
  REAL PRFPAY,LIODAT,COST,NETPAY
  TRANSAC(A,15)=TRANSAC(A-1,15)
  TRANSAC(A,16)=TRANSAC(A-1,16)
  TRANSAC(A,18)=TRANSAC(A-1,18)
  TRANSAC(A,19)=TRANSAC(A,4)
  DO 90 I=1,FLOATD
    IF (CSTFLT(I,1).LE.TRANSAC(A,13)) THEN
      TRANSAC(A,17)=TRANSAC(A,17)+CSTFLT(I,2)
    END IF
90 CONTINUE
  TRANSAC(A,17)=TRANSAC(A,17)+STCOST
  OUTPUT(A,1)=OUTPUT(A-1,1)
  LIODAT=TRANSAC(A,3)*TRANSAC(A,4)
  IF (LIODAT.GT.OUTPUT(A-1,10)) THEN
    LIODAT=OUTPUT(A-1,10)
  END IF
  NETPAY=TRANSAC(A,3)-LIODAT
  IF (TRANSAC(A,10).GT.ALTDY) THEN
    CATCUP=0
    IF (ALQFLG.EQ.0) THEN
      ALQFLG=1
      CATCUP=ALTERF-OUTPUT(A-1,7)
    END IF
  ELSE
    CATCUP=0
  END IF
  COST=TRANSAC(A,3) / (1+PRFRAT)
  COSTPD=COST-TRANSAC(A,18)*COST
  PRFPAY=NETPAY-COSTPD
  IF (ALQFLG.EQ.0) THEN
    ALTERF=ALTERF+(TRANSAC(A,3)-(TRANSAC(A,3)/(1+PRFRAT)))
  END IF
  OUTPUT(A,2)=OUTPUT(A,1)-TRANSAC(A,17)
  OUTPUT(A,3)=TRANSAC(A,17)
  OUTPUT(A,4)=OUTPUT(A-1,4)
  OUTPUT(A,5)=COSTPD+OUTPUT(A-1,5)
  OUTPUT(A,6)=OUTPUT(A,4)+OUTPUT(A,5)
  OUTPUT(A,7)=OUTPUT(A-1,7)+PRFPAY+CATCUP
  OUTPUT(A,8)=OUTPUT(A,6)+OUTPUT(A,7)
  OUTPUT(A,9)=OUTPUT(A,3)-OUTPUT(A,6)
  OUTPUT(A,10)=OUTPUT(A-1,10)-LIODAT+CATCUP
  TOTCHG=TOTCHG+TRANSAC(A,3)
  SUMCST=SUMCST+COST
  I=INT(TODAYS+STDAY)+DELLAG
*****
  ROLLS UP ALL DATA FOR LAST TRANSACTION. (IT IS EXTREMELY
  UNLIKELY THAT AN ACTUAL DELIVERY WOULD BE THE LAST
  TRANSACTION. HOWEVER, IT MAY HAPPEN IF SOMEONE CHOOSES
  TO RUN A COMPLETED CONTRACT TO SEE WHAT THE CONTRACTOR
```

```

INVESTMENT WITH INTEREST ON A MONTHLY BASIS
*****
1- IF INPUT ERROR IN 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 
```

```

*****
      COMPUTES CONTRACTOR INVESTMENT RATIO.
*****
      INVT=FDICOST-INFLOW
      RATIO=INVTD/(FDICOST-COSDEL)
      RETURN
      END
*****      END OF SUBROUTINE INVEST      *****

*****      START OF SUBROUTINE PNTOUT      *****

      SUBROUTINE PNTOUT (TRNFLG,CSTFLG,FRGLG,TRANSAC,ELMENT,FLXFLG,NMTRAN
& ,NMCSLT,STDTE,CSTEL,FRFRAT,STCOST,TOTPYD,TOTLOS,OUTPUT,TOPROF,
& ,PROFIT,FRGRTE,RATIO,FRGLAG,DELLAG,LSTDEL,TITLE,DATHEY,LFLG)
      REAL TRANSAC(25,20),ELMENT(10,5),FRFRAT,TOTLOS,OUTPUT(25,10),
& ,TOPROF,PROFIT,FRGRTE,RATIO,TOTPYD,STCOST
      INTEGER FRGLG,NMCSLT,STDTE,FRGLAG,DELLAG,LSTDEL,PG,NMTRAN
      CHARACTER*3 TRNFLG,CSTFLG,FLXFLG
      CHARACTER*20 COSTEL(10,2)
      CHARACTER*10 ROWA(8),ROWB(10),ROWC(11),ROWD(3),ROWE(10)
      CHARACTER*50 TITLE
      LOGICAL*1 CC
      INTEGER NUM(10),DATHEY
      DATA CC / Z0C /
      OPEN UNIT=1)
      FORMAT( / )
      FORMAT( / )
*****
      ESTABLISHES HEADINGS FOR TRANSACTION REPORT
*****
      PG=0
      ROWA(1)= DATE /
      ROWA(2)= TOTAL
      ROWA(3)= PAID
      ROWA(4)= COST
      ROWA(5)= TOTAL
      ROWA(6)= PROFIT
      ROWA(7)= UNREIM
      ROWA(8)= UNLID
      ROWB(1)= TRANSACT
      ROWB(2)= INCURRED
      ROWB(3)= INCURRED
      ROWB(4)= PROG PAY
      ROWB(5)= PAID AT
      ROWB(6)= COST
      ROWB(7)= PAID AT
      ROWB(8)= TOTAL
      ROWB(9)= PAID
      ROWB(10)= PROGRESS
      ROWC(1)= CODE
      ROWC(2)= COST
      ROWC(3)= FLOAT
      ROWC(4)= COST
      ROWC(5)= RECD
      ROWC(6)= DELIVERY
      ROWC(7)= REIMBURS
      ROWC(8)= DELIVERY
      ROWC(9)= REVENUE

```

```

CAS13790
CAS13800
CAS13810
CAS13820
CAS13830
CAS13840
CAS13850
CAS13860
CAS13870
CAS13880
CAS13890
CAS13900
CAS13910
CAS13920
CAS13930
CAS13940
CAS13950
CAS13960
CAS13970
CAS13980
CAS13990
CAS14000
CAS14010
CAS14020
CAS14030
CAS14040
CAS14050
CAS14060
CAS14070
CAS14080
CAS14090
CAS14100
CAS14110
CAS14120
CAS14130
CAS14140
CAS14150
CAS14160
CAS14170
CAS14180
CAS14190
CAS14200
CAS14210
CAS14220
CAS14230
CAS14240
CAS14250
CAS14260
CAS14270
CAS14280
CAS14290
CAS14300
CAS14310
CAS14320
CAS14330
CAS14340
CAS14350
CAS14360
CAS14370
CAS14380

```

```

      ROWC(10)= COST                                CAS144390
      ROWC(11)= PAYMENTS                            CAS144400
      ROWD(1)= (4+5)                                CAS144410
      ROWD(2)= (6+7)                                CAS144420
      ROWD(3)= (3-6)                                CAS144430
      DO 850 I=1,10                                  CAS144440
        ROWE(I)= (DOLLARS)                          CAS144450
        NUM(I)=I                                    CAS144460
850    CONTINUE                                     CAS144470
100    FORMAT(T43, ' DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL ) CAS144480
110    FORMAT(T51, ' VERSION DATED FEBRUARY 1, 1981 ) CAS144490
120    FORMAT(T57, ' TRANSACTION REPORT )           CAS145000
121    FORMAT(T41,A50)                              CAS14510
122    FORMAT(' ',T13, ' *** UNLIQUIDATED PROGRESS PAYMENTS WOULD HAVE BEEN CAS14520
      &N NEGATIVE WHERE 0 APPEARS IN COLUMN 10. *** ) CAS14530
123    FORMAT(' ',T13, ' *** CHECK PRICES OF DELIVERABLES TO ENSURE CAS14540
      &RE THERE IS NO -FRONT LOADING-. *** )        CAS14550
      COUNT3=34                                     CAS14560
C      ***** CAS14570
C      CONTROLS PRINT ROUTINE WHEN ERRORS ARE DETECTED IN THE CAS14580
C      EDIT SUBROUTINE.                             CAS14590
C      ***** CAS14600
      IF(DADEFY.GT.0) THEN                           CAS14610
        GO TO 1850                                   CAS14620
      END IF                                         CAS14630
C      ***** CAS14640
C      PRINTS TRANSACTION REPORT.                   CAS14650
C      ***** CAS14660
      IF (TRANFLG.EQ. 'YES') THEN                   CAS14670
        DO 1070 I=1,NMTRAN                          CAS14680
          IF (COUNT3.EQ.34) THEN                   CAS14690
            PG=PG+1                                  CAS14700
            WRITE(7,FMT='(A1)' ) CC                 CAS14710
740      FORMAT( / ,T112,A,T113,I2)                 CAS14720
750      FORMAT( / ,A)                               CAS14730
            WRITE(2,480) PG                          CAS14740
            WRITE(2,100)                             CAS14750
            WRITE(2,110)                             CAS14760
            WRITE(2,120)                             CAS14770
            WRITE(2,30)                              CAS14780
            WRITE(2,121) TITLE                       CAS14790
            WRITE(2,*)                                CAS14800
820    FORMAT(' ',T18,I1,T29,I1,T40,I1,T51,I1,T62,I1,T73,I1,T84,I1,T95,I1,CAS14810
      &,T106,I1,T117,I2)                             CAS14820
            WRITE(2,820) (NUM(I),I=1,10)            CAS14830
840    FORMAT(' ',T2,A,T16,A,T38,A,T62,A,T71,A,T81,A,T104,A,T115,A,CAS14840
      &)                                              CAS14850
            WRITE(2,840) (ROWA(I),I=1,8)             CAS14860
860    FORMAT(' ',T1,A,T14,A,T36,A,T47,A,T59,A,T71,A,T81,A,T93,A, CAS14870
      &T104,A,T113,A)                                CAS14880
880    FORMAT(' ',T2,A,T16,A,T27,A,T38,A,T49,A,T52,A,T69,A,T80,A, CAS14890
      &T92,A,T104,A,T113,A)                          CAS14900
            WRITE(2,860) (ROWB(I),I=1,10)            CAS14910
            WRITE(2,890) (ROWC(I),I=1,11)            CAS14920
920    FORMAT(' ',T71,A,T93,A,T104,A)               CAS14930
            WRITE(2,920) ROWD(1),ROWD(2),ROWD(3)     CAS14940
940    FORMAT(' ',T14,A,T25,A,T36,A,T47,A,T58,A,T69,A,T80,A,T91,A,CAS14950
      &T102,A,T113,A)                                CAS14960
            WRITE(2,940) (ROWE(I),I=1,10)            CAS14970
            COUNT3=0                                  CAS14980

```

```

        WRITE(2,200)
        END IF
        FORMAT(      ,I6,      ,I1,I1,I10,I1,I10,I1,I10,I1,I10,I1,
        I10,I1,I10,I1,I10,I1,I10,I1,I10,I1,I10)
        DO 1040 J=1,10
            CALL FUNDB(OUTPUT(I,J))
        CONTINUE
        WRITE(2,990) INT(TRANSAC(I,2)),INT(TRANSAC(I,1)),INT(OUTPUT(I,1),INT(CAS15050
        OUTPUT(I,
        20),INT(OUTPUT(I,3)),INT(OUTPUT(I,4)),INT(OUTPUT(I,5)),INT(OUTPUT(CAS15050
        31,5)),INT(OUTPUT(I,7)),INT(OUTPUT(I,8)),INT(OUTPUT(I,9)),INT(OUTPUT(CAS15050
        31,1,10))
        COUNT3=COUNT3+1
        CONTINUE
        IF (FLAG.EQ.1) THEN
            WRITE(2,20)
            WRITE(2,122)
            WRITE(2,123)
        END IF
    END IF
C *****
C      ESTABLISHES HEADINGS FOR ANALYTICAL REPORT.
C *****
200  FORMAT(TS7,' ANALYTICAL REPORT ')
210  FORMAT(' CONTRACT PRICE ',T31,F10.2)
220  FORMAT(' CONTRACT START DATE ',T35,I6)
230  FORMAT(' START UP COST OF CONTRACT ',T31,F10.2)
240  FORMAT(' DATA FILE PROFIT PERCENTAGE ',T36,F5.2)
250  FORMAT(' COMPLETED PROFIT PERCENTAGE ',T36,F5.2)
260  FORMAT(' PROJECT TITLE ',T18,A50)
270  FORMAT(' TOTAL CONTRACTOR COST ',T31,F10.2)
280  FORMAT('**** PROGRESS PAYMENT RATE DATA****')
290  FORMAT(' PROGRESS PAYMENT RATE ',T35,' * ',T38,F6.2,T45,
& ' * ')
300  FORMAT(' PROGRESS PAYMENT RATE ',T35,' * ',T38,F6.2,T45, ' * ',T68,
& ' * ' THESE VALUES REFLECT COMPUTATIONS *')
310  FORMAT(' ORDINARY LIQUIDATION RATE ',T35,' * ',T38,F6.2,T45, ' * ')
320  FORMAT(' ORDINARY LIQUIDATION RATE ',T35,' * ',T38,F6.2,T45, ' * ',T68,CAS15050
& ' * ' BASED ON OVERLIQUIDATION OF *')
330  FORMAT(' ALTERNATE LIQUIDATION RATE ',T35,' * ',T38,F6.2,T45, ' * ')
340  FORMAT(' ALTERNATE LIQUIDATION RATE ',T35,' * ',T38,F6.2,T45, ' * ',
& T68, ' * ' PROGRESS PAYMENTS. SEE NOTE AT *')
350  FORMAT(' CONTRACTOR INVESTMENT RATIO ',T35,' * ',T39,F5.2,T45, ' * ')
360  FORMAT(' CONTRACTOR INVESTMENT RATIO ',T35,' * ',T39,F5.2,T45, ' * ',
& T68, ' * ' END OF TRANSACTION REPORT. *')
370  FORMAT(' *****
& *****')
380  FORMAT(' PROGRESS PAYMENT LAG TIME IN DAYS ',T41,I3)
390  FORMAT(' DELIVERY PAYMENT LAG TIME IN DAYS ',T41,I3)
400  FORMAT(' DATE OF FINAL DELIVERY ',T40,I6)
410  FORMAT(' PROGRESS PAYMENT RATE COMPUTED BY PROGRAM ',T34,F6.2)
420  FORMAT(' PROGRESS PAYMENT RATE INPUT BY USER INTO PROGRAM ',T34,
& F6.2)
430  FORMAT(T31, ' COST ELEMENT DATA ')
440  FORMAT(' # ,I6, NAME ',T28, UNPAID COST ,T41, %PAID ,T48, PAID
& PAID ,T60, ELIGIBILITY )
450  FORMAT('I6, DAYS FLOAT ,T48, DAYS FLOAT ,T60, PROGRESS PAYMENT
&
460  FORMAT(' ,I2,T6,H20,T31,I3,T41,F5.1,T51,I3,I6, SUBCONTRACT PROGRESS
&

```



|     |   |          |
|-----|---|----------|
| 440 | FORMAT(' ',I2,T6,A20,T31,I3,T41,F5.1,T51,I3,T60,' PAID COST') | CAS15590 |
| 450 | FORMAT(' ',I2,T6,A20,T31,I3,T60,' INCURRED COST')             | CAS15600 |
| 460 | FORMAT(' ',I2,T6,A20,T51,I3,T60,' INCURRED COST')             | CAS15610 |
| 470 | FORMAT(T35,' *****')  | CAS15620 |
| 480 | FORMAT(T35,' *****',T68,' *****')                             | CAS15630 |
| 490 | FORMAT(T123,' PAGE ',T128,I3)                                 | CAS15640 |
| 495 | FORMAT(' PROFIT PAID TO CONTRACTOR',T31,F10.2)                | CAS15650 |
| 500 | FORMAT(T35,' *',T45,' *')                                     | CAS15660 |
| 505 | FORMAT(T35,' *',T45,' *',T68,' *',T108,' *')                  | CAS15670 |
|     | WRITE(2,FMT='(A1)') CC  | CAS15680 |
| C   | *****   | CAS15690 |
| C   | PRINTS SUMMARY DATA IN ANALYTICAL REPORT.                     | CAS15700 |
| C   | *****   | CAS15710 |
|     | PG=PG+1   | CAS15720 |
|     | WRITE(2,480) PG   | CAS15730 |
|     | WRITE(2,30)   | CAS15740 |
|     | WRITE(2,100)  | CAS15750 |
|     | WRITE(2,110)  | CAS15760 |
|     | WRITE(2,200)  | CAS15770 |
|     | WRITE(2,30)   | CAS15780 |
|     | WRITE(2,260) TITLE  | CAS15790 |
|     | WRITE(2,30)   | CAS15800 |
|     | WRITE(2,330)  | CAS15810 |
|     | WRITE(2,210) TOTPYD   | CAS15820 |
|     | WRITE(2,270) TOTCOS   | CAS15830 |
|     | WRITE(2,490) TOPROF   | CAS15840 |
|     | WRITE(2,230) STCOST   | CAS15850 |
|     | WRITE(2,220) STDATE   | CAS15860 |
|     | PRFRAT=PRFRAT*100   | CAS15870 |
|     | PROFIT=PROFIT*100   | CAS15880 |
|     | WRITE(2,240) PRFRAT   | CAS15890 |
|     | WRITE(2,250) PROFIT   | CAS15900 |
|     | WRITE(2,330)  | CAS15910 |
|     | WRITE(2,20)   | CAS15920 |
|     | WRITE(2,280)  | CAS15930 |
|     | WRITE(2,20)   | CAS15940 |
|     | PRGRTE=PRGRTE*100   | CAS15950 |
|     | RATIO=RATIO*100   | CAS15960 |
| C   | *****   | CAS15970 |
| C   | PRINTS PROGRESS PAYMENT RATE, ORDINARY AND ALTERNATE          | CAS15980 |
| C   | LIQUIDATION RATES, AND CONTRACTOR INVESTMENT RATIO.           | CAS15990 |
| C   | ALSO PRINTS WARNING IF FLEX RATE IS COMPUTED BASED ON         | CAS16000 |
| C   | OVERLIQUIDATION OF PROGRESS PAYMENTS.                         | CAS16010 |
| C   | *****   | CAS16020 |
|     | IF (LFLG.EQ.1) THEN   | CAS16030 |
|     | WRITE(2,475)  | CAS16040 |
|     | WRITE(2,295) PRGRTE   | CAS16050 |
|     | WRITE(2,505)  | CAS16060 |
|     | WRITE(2,305) PRGRTE   | CAS16070 |
|     | WRITE(2,505)  | CAS16080 |
|     | WRITE(2,315) TRNSAC(NMTRAN,19)*100                            | CAS16090 |
|     | WRITE(2,505)  | CAS16100 |
|     | WRITE(2,325) RATIO  | CAS16110 |
|     | WRITE(2,475)  | CAS16120 |
|     | ELSE  | CAS16130 |
|     | WRITE(2,470)  | CAS16140 |
|     | WRITE(2,290) PRGRTE   | CAS16150 |
|     | WRITE(2,500)  | CAS16160 |
|     | WRITE(2,300) PRGRTE   | CAS16170 |

```

        WRITE(2,500)
        WRITE(2,310) TRNSAC(NMTRAN,19)*100
        WRITE(2,500)
        WRITE(2,320) RATIO
        WRITE(2,470)
    END IF
    WRITE(2,30)
    WRITE(2,330)
    IF (PRGFLG.EQ.0) THEN
510      FORMAT(' ALL PROGRESS PAYMENTS INPUT ARE ACTUAL. ')
        WRITE(2,510)
    END IF
C      *****
C      PRINTS COST ELEMENT DATA.
C      *****
    WRITE(2,330)
    WRITE(2,20)
    WRITE(2,400)
    WRITE(2,410)
    WRITE(2,420)
    WRITE(2,330)
    DO 1820 I=1,NMCSTL
        IF (ELMENT(I,1).EQ.1.AND.ELMENT(I,2).GE.0) THEN
            WRITE(2,450) I,COSTEL(I,1),INT(ELMENT(I,2))
        END IF
        IF (ELMENT(I,1).EQ.1.AND.ELMENT(I,2).LT.0) THEN
            WRITE(2,460) I,COSTEL(I,1),INT(ELMENT(I,2))
        END IF
        ELMENT(I,3)=ELMENT(I,3)*100
        IF (ELMENT(I,1).EQ.2) THEN
            WRITE(2,440) I,COSTEL(I,1),INT(ELMENT(I,2)),ELMENT(I,3),
&INT(ELMENT(I,4))
        END IF
        IF (ELMENT(I,1).EQ.3) THEN
            WRITE(2,430) I,COSTEL(I,1),INT(ELMENT(I,2)),ELMENT(I,3),
&INT(ELMENT(I,4))
        END IF
1820 CONTINUE
    WRITE(2,330)
    WRITE(2,340) PRGLAG
    WRITE(2,350) DELLAG
    WRITE(2,360) LSTDEL
    WRITE(2,20)
    IF (FLXFLG.EQ.'YES') THEN
        WRITE(2,390) PRGRTE
    ELSE
        WRITE(2,380) PRGRTE
    END IF
    WRITE(2,330)
C      *****
C      PRINTS COST RECAP REPORT.
C      *****
1150 IF (DATHEY.EQ.1.OR.DATHEY.EQ.3) THEN
        COSTFLG='YES'
    END IF
    IF (COSTFLG.EQ.'YES') THEN
        TRNSAC(1,14)=TRNSAC(1,14)+STCOST
        COUNT3=36
    DO 2190 I=1,NMTRAN
        IF (COUNT3.EQ.36) THEN

```

CAS16190  
 CAS16200  
 CAS16210  
 CAS16220  
 CAS16230  
 CAS16240  
 CAS16250  
 CAS16260  
 CAS16270  
 CAS16280  
 CAS16290  
 CAS16300  
 CAS16310  
 CAS16320  
 CAS16330  
 CAS16340  
 CAS16350  
 CAS16360  
 CAS16370  
 CAS16380  
 CAS16390  
 CAS16400  
 CAS16410  
 CAS16420  
 CAS16430  
 CAS16440  
 CAS16450  
 CAS16460  
 CAS16470  
 CAS16480  
 CAS16490  
 CAS16500  
 CAS16510  
 CAS16520  
 CAS16530  
 CAS16540  
 CAS16550  
 CAS16560  
 CAS16570  
 CAS16580  
 CAS16590  
 CAS16600  
 CAS16610  
 CAS16620  
 CAS16630  
 CAS16640  
 CAS16650  
 CAS16660  
 CAS16670  
 CAS16680  
 CAS16690  
 CAS16700  
 CAS16710  
 CAS16720  
 CAS16730  
 CAS16740  
 CAS16750  
 CAS16760  
 CAS16770  
 CAS16780

```

COUNT3=0
WRITE(2,FMT='(A1)' ) CC
PG=PG+1
WRITE(2,480) PG
1900 FORMAT(T54,' RECAP OF COST STATEMENT ')
1905 FORMAT(T36,' NOTE: TOTAL FOR FIRST COST STATEMENT INCLUDES START UCAS16840
&P COST ')
WRITE(2,100)
WRITE(2,110)
WRITE(2,1900)
WRITE(2,1905)
WRITE(2,30)
WRITE(2,121) TITLE
WRITE(2,30)
2060 FORMAT(' ',T2,' DATE',T17,' #1',T28,' #2',T39,' #3',T50,' #4',T61,CAS16850
&' #5',T72,' #6',T83,' #7',T95,' #8',T105,' #9',T114,' #10',T122,CAS16860
&' TOTAL')
WRITE(2,2060)
END IF
2110 FORMAT(' ',1X,I6,1X,11(I10,1X))
2120 FORMAT(' ',1X,' TOTAL',1X,11(I10,1X))
IF (TRANSAC(I,1).EQ.3) THEN
WRITE(2,2110) INT(TRANSAC(I,2)),INT(TRANSAC(I,3)),INT(TRANSAC(I,CAS16870
&4)),INT(TRANSAC(I,5)),INT(TRANSAC(I,6)),INT(TRANSAC(I,7)),INT(TRANSAC(CAS16880
&I,8)),INT(TRANSAC(I,9)),INT(TRANSAC(I,10)),INT(TRANSAC(I,11)),INT(TRANSAC(CAS16890
&SAC(I,12)),INT(TRANSAC(I,14))
COUNT3=COUNT3+1
END IF
2150 CONTINUE
WRITE(2,20)
WRITE(2,2120) INT(ELMENT(1,5)),INT(ELMENT(2,5)),INT(ELMENT(CAS16900
&3,5)),INT(ELMENT(4,5)),INT(ELMENT(5,5)),INT(ELMENT(6,5)),INT(ELMENT(CAS17000
&T(7,5)),INT(ELMENT(8,5)),INT(ELMENT(9,5)),INT(ELMENT(10,5)),INT(TOCAS17100
&TOS)
END IF
*****
C PRINTS DELIVERY SUMMARY.
C *****
IF (DATKEY.EQ.2.OR.DATKEY.EQ.3) THEN
COUNT3=36
DO 2200 I=1,NMTRAN
IF (COUNT3.EQ.36) THEN
COUNT3=0
WRITE(2,FMT='(A1)' ) CC
PG=PG+1
WRITE(2,480) PG
WRITE(2,100)
WRITE(2,110)
WRITE(2,2300)
WRITE(2,30)
WRITE(2,121) TITLE
WRITE(2,30)
WRITE(2,2310)
WRITE(2,2320)
END IF
IF (TRANSAC(I,1).EQ.4.OR.TRANSAC(I,1).EQ.5) THEN
WRITE(2,2300) INT(TRANSAC(I,2)),INT(TRANSAC(I,3))
COUNT3=COUNT3+1
END IF
2200 CONTINUE

```

|   |          |
|---|----------|
| WRITE(2,20)   | CAS17280 |
| WRITE(2,2350) INT(TOTFYD)   | CAS17400 |
| END IF  | CAS17410 |
| 200 FORMAT(T58, DELIVERY SUMMARY)                                     | CAS17420 |
| 210 FORMAT(T49, TRANSACTION, T75, DELIVERY)                           | CAS17430 |
| 220 FORMAT(T52, DATE, T76, AMOUNT)                                    | CAS17440 |
| 230 FORMAT(T52, T16, T76, T16)  | CAS17450 |
| 250 FORMAT(T52, TOTAL, T74, T18)                                      | CAS17460 |
| CLOSE(UNIT=2)   | CAS17470 |
| RETURN  | CAS17480 |
| END   | CAS17490 |
| ***** END OF SUBROUTINE PNTOUT *****                                  | CAS17500 |
|   | CAS17510 |
|   | CAS17520 |
|   | CAS17530 |
| ***** START OF SUBROUTINE SORT *****                                  | CAS17540 |
|   | CAS17550 |
| ***** SORTS TRANSACTIONS AFTER PLANNED PROGRESS PAYMENT TRANSACTIONS/ | CAS17560 |
| ***** DATES AND DATES FOR RECEIPT OF DELIVERY PAYMENTS ARE CREATED.   | CAS17570 |
|   | CAS17580 |
| SUBROUTINE SORT(ARNAME,ROWS,COLMNS,SORTCM)                            | CAS17590 |
| REAL TEMP   | CAS17600 |
| INTEGER ROWS,COLMNS,SORTCM  | CAS17610 |
| DIMENSION ARNAME(325,20)  | CAS17620 |
| DO 140 I=1,ROWS   | CAS17630 |
| DO 120 J=1,(ROWS-1)   | CAS17640 |
| IF (ARNAME(J+1,SORTCM).LT.ARNAME(J,SORTCM)) THEN                      | CAS17650 |
| DO 100 K=1,COLMNS   | CAS17660 |
| TEMP=ARNAME(J,K)  | CAS17670 |
| ARNAME(J,K)=ARNAME(J+1,K)   | CAS17680 |
| ARNAME(J+1,K)=TEMP  | CAS17690 |
| 100 CONTINUE  | CAS17700 |
| END IF  | CAS17710 |
| 120 CONTINUE  | CAS17720 |
| 140 CONTINUE  | CAS17730 |
| RETURN  | CAS17740 |
| END   | CAS17750 |
| ***** END OF SUBROUTINE SORT *****                                    | CAS17760 |
|   | CAS17770 |
|   | CAS17780 |
|   | CAS17790 |
| ***** START OF SUBROUTINE YRMODA *****                                | CAS17800 |
|   | CAS17810 |
| ***** USED WITH FDAY TO CONVERT YYMMDD DATES TO DAYS AFTER AWARD      | CAS17820 |
|   | CAS17830 |
| SUBROUTINE YRMODA(YR,MO,DA,J)   | CAS17840 |
| INTEGER YR,DA   | CAS17850 |
| IF (MO.GE.3) GO TO 1  | CAS17860 |
| M=MO+9  | CAS17870 |
| J=YR-1961   | CAS17880 |
| GO TO 2   | CAS17890 |
| 1 M=MO+3  | CAS17900 |
| J=YR-1960   | CAS17910 |
| 2 J=J+1461  | CAS17920 |
| IF (J.LT.0) J=J+3   | CAS17930 |
| J=J+4+(150*M+2)/5+DA  | CAS17940 |
| RETURN  | CAS17950 |
| END   | CAS17960 |
| ***** END OF SUBROUTINE YRMODA *****                                  | CAS17970 |
|   | CAS17980 |

```

C ***** START OF SUBROUTINE ROUND *****
C ***** ROUNDS OUTPUT MATRIX PRIOR TO PRINTING TRANSACTION REPORT *****
C
SUBROUTINE ROUND(IRD)
REAL IRD,IFL
  IIT=INT(IRD)
  IFL=IRD-IIT
  IF (IFL.GE. (.5)) THEN
    IRD=IRD+1
    GOTO 13520
  END IF
  IF (IFL*(-1).GE. (.5)) THEN
    IRD=IRD+(-1)
  END IF
13520 RETURN
END
C ***** END OF SUBROUTINE ROUND *****
C
C ***** START OF INTEGER FUNCTION FDAY *****
C ***** USED WITH YRMODA TO CONVERT YYMMDD DATES TO DAYS AFTER AWARD *****
C
INTEGER FUNCTION FDAY(IN)
INTEGER IYR,MO,IDA,IN
  IYR=IN / 10000
  I=IYR*10000
  J=IN-I
  MO=J / 100
  I=MO*100
  IDA=J-I
  CALL YRMODA(1900+IYR,MO,IDA,I)
  FDAY=I
  RETURN
C ***** USED WITH JYMD TO CONVERT DAYS AFTER AWARD TO YYMMDD DATES *****
C
ENTRY FDATE(IN)
  CALL JYMD(IN,IYR,MO,IDA)
  IYR=(IYR-1900)*10000
  MO=MO*100
  FDATE=IYR+MO+IDA
  RETURN
END
C ***** END OF INTEGER FUNCTION FDAY *****
C
C ***** START OF SUBROUTINE JYMD *****
C ***** USED WITH FDATE TO CONVERT DAYS AFTER AWARD TO YYMMDD DATES *****
C
SUBROUTINE JYMD(J,YR,MO,DA)
INTEGER YR,DA
  IF (J.GT.0) GOTO 4
  YR=4-J / 1461-1
  DA=J-(YR*1461-7) / 4
  GOTO 3

```

```

CAS18370
CAS18380
CAS18390
CAS18400
CAS18410
CAS18420
CAS18430
CAS18440
CAS18450
CAS18460
CAS18470
CAS18480
CAS18490
CAS18500
CAS18510
CAS18520
CAS18530
CAS18540
CAS18550
CAS18560
CAS18570
CAS18580
CAS18590
CAS18600
CAS18610
CAS18620
CAS18630
CAS18640
CAS18650
CAS18660
CAS18670
CAS18680
CAS18690
CAS18700
CAS18710
CAS18720
CAS18730
CAS18740
CAS18750
CAS18760
CAS18770
CAS18780
CAS18790
CAS18800
CAS18810
CAS18820
CAS18830
CAS18840
CAS18850
CAS18860
CAS18870
CAS18880
CAS18890
CAS18900
CAS18910
CAS18920
CAS18930
CAS18940
CAS18950
CAS18960
CAS18970
CAS18980
CAS18990
CAS19000

```

|                |   |          |
|----------------|---|----------|
| 4              | YR=(4*J-1) / 1461   | CAS18590 |
|                | DA=J-(YR*1461) / 4  | CAS18590 |
|                | MO=(5*DA-3) / 153   | CAS18510 |
|                | DA=DA-(153*MO+2) / 5  | CAS18620 |
|                | IF(MO-9) 2,2,1  | CAS18620 |
|                | MO=MO+3   | CAS18640 |
|                | YR=YR+1960  | CAS18650 |
|                | RETURN  | CAS18660 |
| 1              | MO=MO-9   | CAS18670 |
|                | YR=YR+1961  | CAS18680 |
|                | RETURN  | CAS18690 |
|                | END   | CAS18700 |
| IF(LESS OPT(3) |   | CAS18710 |
|                | SUBROUTINE GETSTR(STRNG,DELIM,SUBSTR,IFOS)                      | CAS18720 |
|                | CHARACTER*(*) STRNG,SUBSTR,DELIM                                | CAS18730 |
|                | INTEGER*4 IFOS,LENDLM,LENSTR                                    | CAS18740 |
|                | IFOS=INDEX(STRNG,DELIM)   | CAS18750 |
|                | IF (IFOS.GT.0) THEN   | CAS18760 |
|                | LENDLM=LEN(DELIM)   | CAS18770 |
|                | LENSTR=LEN(STRNG)   | CAS18780 |
|                | IF (IFOS-1.LE.0) THEN   | CAS18790 |
|                | SUBSTR = ' '  | CAS18800 |
|                | GOTO 100  | CAS18810 |
|                | ENDIF   | CAS18820 |
|                | SUBSTR=STRNG(1:IFOS-1)  | CAS18830 |
| 100            | STRNG=STRNG(IFOS+LENDLM:LENSTR)                                 | CAS18840 |
|                | ELSE  | CAS18850 |
|                | SUBSTR=STRNG  | CAS18860 |
|                | STRNG=' '   | CAS18870 |
|                | ENDIF   | CAS18880 |
|                | RETURN  | CAS18890 |
|                | END   | CAS18900 |
| IF(LESS OPT(3) |   | CAS18910 |
|                | FUNCTION REALSV(STRNG1,IN)                                      | CAS18920 |
|                | CHARACTER*8 STRNG1,STRNG2                                       | CAS18930 |
|                | CHARACTER*1 STRARY(8)   | CAS18940 |
|                | INTEGER*4 I,J,IDECP,INTLN,ISIGN,ISTRN,IN,IDECE(8),IPWR          | CAS18950 |
|                | INTEGER*4 IBLNK,IBLNK2  | CAS18960 |
|                | REAL REALSV   | CAS18970 |
|                | EQUIVALENCE (STRNG2,STRARY)                                     | CAS18980 |
|                | STRNG2=STRNG1   | CAS18990 |
|                | J=1   | CAS19000 |
|                | IBLNK=0   | CAS19010 |
|                | IBLNK2=0  | CAS19020 |
|                | IN=0  | CAS19030 |
|                | IDECP=0   | CAS19040 |
|                | ISIGN=1   | CAS19050 |
|                | REALSV=0  | CAS19060 |
|                | DO 100 I=1,8,1  | CAS19070 |
|                | IDECE(I)= ICHAR(STRARY(I))                                      | CAS19080 |
|                | IF ((IDECE(I).LT.240.OR.IDECE(I).GT.249).AND.(IDECE(I).NE.75.AN | CAS19090 |
|                | 10. IDECE(I).NE.96.AND. IDECE(I).NE.54)) THEN                   | CAS19100 |
|                | IN=1  | CAS19110 |
|                | GOTO 400  | CAS19120 |
|                | ENDIF   | CAS19130 |
|                | IF (IDECE(I).EQ.75.AND. IDECP.NE.0) THEN                        | CAS19140 |
|                | IN=1  | CAS19150 |
|                | GOTO 400  | CAS19160 |
|                | ENDIF   | CAS19170 |
|                | IF (IDECE(I).EQ.75) IDECP=I                                     | CAS19180 |

```

IF (IDECED(I).EQ.96.AND.I.NE.1) THEN
    IN=1
    GOTO 400
ENDIF
IF (IDECED(I).EQ.96) ISIGN=-1
IF (IDECED(I).EQ.64.AND.IBLNF.EQ.1.AND.IBLNF2.EQ.1) GOTO 100
IF (IDECED(I).EQ.64.AND.IBLNF.EQ.1.AND.IBLNF2.EQ.0) THEN
    IN=1
    GOTO 400
ENDIF
IF (IDECED(I).EQ.64.AND.IBLNF.EQ.0) THEN
    IBLNF=1
    IN=1
    IBLNF2=1
    GOTO 100
ENDIF
IBLNF2=0
IN=I+1
100 CONTINUE
IF (ISIGN.EQ.-1.AND.IN.EQ.2) THEN
    IN=1
    GOTO 400
ENDIF
IF (IDECFL.GT.0) THEN
    INTLN=IDECFL-1
    ISTRLN=IN-1
ELSE
    INTLN=IN-1
    ISTRLN=INTLN
ENDIF
IF (ISIGN.EQ.-1) THEN
    J=2
    INTLN=INTLN-1
ENDIF
IFWR=INTLN-1
DO 300 I=J,ISTRLN,1
IF (IDECED(I).EQ.75) THEN
    GOTO 300
ENDIF
REALSV=(IDECED(I)-240.)*(10.** (IFWR))+REALSV
IFWR=IFWR-1
300 CONTINUE
REALSV=REALSV*ISIGN
400 RETURN
END
*PROCESS OFT(3)
FUNCTION INTSTV(STRNG1,IN)
CHARACTER*(*) STRNG1
CHARACTER*8 STRNG2
CHARACTER*1 STRARY(8)
INTEGER*4 I,J,IN,IDECED(5),IFWR,ISIGN,IBLNF,IBLNF2
EQUIVALENCE (STRNG2,STRARY)
STRNG2=STRNG1
J=1
IBLNF=0
IBLNF2=0
IN=0
ISIGN=1
INTLN=0
DO 100 I=1,8,1

```

```

CAS19190
CAS19200
CAS19210
CAS19220
CAS19230
CAS19240
CAS19250
CAS19260
CAS19270
CAS19280
CAS19290
CAS19300
CAS19310
CAS19320
CAS19330
CAS19340
CAS19350
CAS19360
CAS19370
CAS19380
CAS19390
CAS19400
CAS19410
CAS19420
CAS19430
CAS19440
CAS19450
CAS19460
CAS19470
CAS19480
CAS19490
CAS19500
CAS19510
CAS19520
CAS19530
CAS19540
CAS19550
CAS19560
CAS19570
CAS19580
CAS19590
CAS19600
CAS19610
CAS19620
CAS19630
CAS19640
CAS19650
CAS19660
CAS19670
CAS19680
CAS19690
CAS19700
CAS19710
CAS19720
CAS19730
CAS19740
CAS19750
CAS19760
CAS19770
CAS19780
CAS19790
CAS19800
CAS19810
CAS19820
CAS19830
CAS19840
CAS19850
CAS19860
CAS19870
CAS19880
CAS19890
CAS19900

```

|     |   |          |
|-----|---|----------|
|     | IDECEQ(I)= ICHAR(STARRY(I))                                       | CAS19790 |
|     | IF (IDECEQ(I).LT.240.OR.IDECEQ(I).GT.249).AND.IDECEQ(I).NE.96.AND | CAS19800 |
|     | 1. IDECEQ(I).NE.64) THEN  | CAS19810 |
|     | IN=1  | CAS19820 |
|     | GOTO 400  | CAS19830 |
|     | ENDIF   | CAS19840 |
|     | IF (IDECEQ(I).EQ.96.AND.1.NE.1) THEN                              | CAS19850 |
|     | IN=1  | CAS19860 |
|     | GOTO 400  | CAS19870 |
|     | ENDIF   | CAS19880 |
|     | IF (IDECEQ(I).EQ.96) ISIGN= -1                                    | CAS19890 |
|     | IF (IDECEQ(I).EQ.64.AND. IBLNF.EQ.1.AND. IBLNF 2.EQ.1) GOTO 100   | CAS19900 |
|     | IF (IDECEQ(I).EQ.64.AND. IBLNF.EQ.1.AND. IBLNF 2.EQ.0) THEN       | CAS19910 |
|     | IN=1  | CAS19920 |
|     | GOTO 400  | CAS19930 |
|     | ENDIF   | CAS19940 |
|     | IF (IDECEQ(I).EQ.64.AND. IBLNF.EQ.0) THEN                         | CAS19950 |
|     | IBLNF=1   | CAS19960 |
|     | IN=1  | CAS19970 |
|     | IBLNF 2=1   | CAS19980 |
|     | GOTO 100  | CAS19990 |
|     | ENDIF   | CAS20000 |
|     | IBLNF 2=0   | CAS20010 |
|     | IN=I+1  | CAS20020 |
| 100 | CONTINUE  | CAS20030 |
|     | IF (ISIGN.EQ.-1.AND. IN.EQ.2) THEN                                | CAS20040 |
|     | IN=1  | CAS20050 |
|     | GOTO 400  | CAS20060 |
|     | ENDIF   | CAS20070 |
|     | ISTRLN=IN-1   | CAS20080 |
|     | IF (ISIGN.EQ.-1) THEN   | CAS20090 |
|     | J=2   | CAS20100 |
|     | IFWR=ISTRLN-2   | CAS20110 |
|     | ELSE  | CAS20120 |
|     | IFWR=ISTRLN-1   | CAS20130 |
|     | ENDIF   | CAS20140 |
|     | DO 300 I=J,ISTRLN,1   | CAS20150 |
|     | INTSTV=(IDECEQ(I)-240)*(10**IFWR)+INTSTV                          | CAS20160 |
| 200 | IFWR=IFWR-1   | CAS20170 |
|     | INTSTV=INTSTV*ISIGN   | CAS20180 |
| 400 | RETURN  | CAS20190 |
|     | END   | CAS20200 |

F;

G



**DATE**  
**ILMEI**